Master of Science in Computer Science
June 2020


# BLE Beacon Based Indoor Positioning System in an Office Building using Machine Learning

**Rohan Reddy Tirumalareddy**

Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden

This thesis is submitted to the Faculty of Computing at Blekinge Institute of Technology in partial fulfilment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full time studies.

The author declares that they are the sole author of this thesis and that they have not used any sources other than those listed in the bibliography and identified as references. They further declare that they have not submitted this thesis at any other institution to obtain a degree.

**Contact Information:**
Author(s):
Rohan Reddy Tirumalareddy
E-mail: tiro17@student.bth.se

E-mail: poulrohan23@gmail.com

University advisor:
Prof. Yulia Sidorova
Department of Computer Science

Faculty of Computing
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

Internet : www.bth.se
Phone : +46 455 38 50 00
Fax : +46 455 38 50 57

# Abstract

**Context:** Indoor positioning systems have become more widespread over the past decade, mainly due to devices such as Bluetooth Low Energy beacons which are low at cost and work effectively. The context of this thesis is to localize and help people navigate to the office equipment, meeting rooms, etc., in an office environment using machine learning algorithms. This can help the employees to work more effectively and conveniently saving time.
**Objective:** To perform a literature review of various machine learning models in indoor positioning that are suitable for an office environment. Also, to experiment with those selected models and compare the results based on their performance. An android smartphone and BLE beacons have been used to collect RSSI values along with their respective location coordinates for the dataset. Besides, the accuracy of positioning is determined by using state-of-the-art machine learning algorithms to train the dataset. Using performance metrics such as Euclidean distance error, CDF curve of Euclidean distance error, RMSE and MAE to compare results and select the best model for this research.
**Methods:** A Fingerprinting method for indoor positioning is studied and applied for the collection of the RSSI values and (x, y) location coordinates from the fixed beacons. A literature review is performed on various machine learning models appropriate for indoor positioning. The chosen models were experimented and compared based on their performances using performance metrics such as CDF curve, MAE, RSME and Euclidean distance error.
**Results:** The literature study shows that Long Short Term Memory and Multi-layer Perceptron, Gradient boosting, XG boosting and Ada boosting are suitable models for indoor positioning. The experimentation and comparison of these models show that the overall performance of Long short-term memory network was better than Multilayer Perceptron, Gradient boosting, XG boosting and Adaboosting.
**Conclusions:** After analysing the acquired results and taking into account the real-world scenarios to which this thesis is intended, it can be stated that the LSTM network provides the most accurate location estimation using beacons. This system can be monitored in real-time for maintenance and personnel tracking in an office environment.

**Keywords:** Indoor Positioning System, Fingerprinting, Machine Learning, Bluetooth Low Energy

# Acknowledgments

I would like to thank Professor Yulia Sidorova for giving me suggestions and constructive criticism. Her enormous support ranged from leading me in the right direction to correcting the choice of words.

This thesis was supported by NavAlarm, Linköping. I would like to thank Sara M. Razavi for offering me the chance to do my Master's thesis in her office and to share her knowledge and skills with me throughout the thesis. Thanks go to my colleagues at NavAlarm for supporting and helping me, especially Sidharth and Sanjana. Additionally, I'm grateful and blessed to have supportive parents and friends.

# Contents

# List of Figures

# List of Tables

# List of Abbreviations

| | |
|---|---|
| IPS | Indoor Positioning System |
| GPS | Global Positioning System |
| BLE | Bluetooth Low Energy |
| LSTM | Long Short-Term Memory |
| MLP | Multi Layer Perceptron |
| RFID | Radio-Frequency Identification |
| UWB | Ultra Wide Band |
| RSSI | Received Signal Strength Indication |
| LOS | Line Of Sight |
| NLOS | Non - Line of Sight |
| RMSE | Root Mean Squared Error |
| MAE | Mean Absolute Error |
| CDF | Cumulative Distribution Function |
| RSS | Received Signal Strength |
| TOA | Time of Arrival |
| TDOA | Time Difference of Arrival |
| WLAN | Wireless Local Area Network |
| NFC | Near-Field Communication |
| QR | Quick Response |
| UUID | Universally Unique Identifier |
| API | Application Programming Interface |
| SDK | Software Development Kit |
| m | meters |

# Chapter 1

# Introduction

In recent years, location-based services have been widely implemented in various applications and services in mobile environments, where location awareness is crucial to people in their daily lives [13]. For an emergency evacuation of office personnel, response teams need positioning solutions to assist evacuation and rescue operations [14][15][16]. In an Office environment, location-aware service such as on-demand lighting or ventilation control, meeting rooms and unutilized spaces should be known for occupants [17][18]. TThere is a need for a navigation system as GPS, but which works in indoors, that helps indoor pedestrians to reach to their destination [19][20].

The GPS satellites are widely used by people in everyday life, but this type of positioning only navigates in outdoors or open spaces [21]. GPS is a type of technology that is not intended for indoor navigation and localization applications [22]. In order to overcome the drawback of GPS, there has been a lot of studies conducted to build navigation and positioning indoors, based on these studies people have developed different methods in positioning such as WiFi, Radio-Frequency Identification detector (RFID) and Bluetooth low energy (BLE) beacon technology [23]. To enable smart services for energy saving in a building, location-aware technology can be adapted to locate persons or devices and provide an appropriate service. WiFi, RFID and BLE beacons are thus a potentially viable solution [24]. Where WiFi and RFID, have some inevitable disadvantages such as WiFi is dependent on power supply and stationary transmitter does not move from one place to another, while RFID despite its low cost, does not have enough bandwidth for large scale and is restricted for users with special equipment as well as it works only in close contact or if more receivers are present [21][23].The location sensing information in indoor environments requires higher precision and is a more challenging task in part because various obstacles or people are present. The UWB will not work in these situations were objects are present, as it only works well in empty spaces[25].

The BLE beacon technology, a subset of Bluetooth technology which is capable of broadcasting signals using a nominal amount of battery power [26]. BLE is a new high potential technology with low energy consumption and low cost. Using a button battery, a Bluetooth beacon can stay active from six months up to two years depending on the frequency and its own power usage [23]. These beacons can be distinguished by a unique identification number of each beacon. Received Signal Strength Indicator (RSSI) values are used to estimate the approximate distance between the receiver and the beacon, the RSSI shows no signal when the receiving

smartphone device moves far from the beacon [27].

The indoor positioning algorithm is roughly classified into two types: triangular positioning and fingerprint positioning [28][29]. Triangular positioning reduces considerably as it relies heavily on the path model's rationality but focuses only on the condition of the LOS, it does not concentrate on the situation of the NLOS and bad positioning [30]. On the other hand, the fingerprint approach has two phases [31]. The first is the offline phase, the location fingerprints are collected by dividing the location into rectangular grids, and multiple access points are fixed to collect the RSSI values at each grid location to train machine learning model [11] [31]. The second is online phase, there are two phases in online: data collection and analysis [11]. The Other method which can be used for IPS is Trilateration, it works by finding the intersection point between the signal strength. This method is most commonly used in outdoor navigation and GPS. But the main disadvantage of this method is that it only prefers LOS conditions. Any obstacles in the environment can affect its signal strength and delivers poor accuracy [5].

In this paper, the proposed protocol improves the existing fingerprinting technique using machine-learning algorithms such as LSTM, MLP, Gradient boosting, XG boosting, Adaboosting. The proposed system shall predict the user's current location based on the input from the data received from the beacons to the mobile application in the form of RSSI values [32]. An experiment is conducted to analyse and evaluate these different machine learning models and are trained with the RSSI sample values at various radio mapping points [23][30][21][33]. Once the system is trained, the machine estimates a mobile's position based on the input given by the RSSI values. The proposed system utilizes the popular state-of-the-art machines-learning technique for both fingerprinting and location estimation [11].The experiment shows the working of these different machine learning models and their respective predicted locations. The data used for this thesis consists of (x,y) location coordinates from different radio mapping points with their RSSI values received from BLE beacons which are spread across the floor plan.

## 1.1 Problem Description

Office environment is subjected to a lot of movement from office personnel and equipment. It is often challenging to keep track of the people who are struck inside the building at the time of an emergency evacuation. Even at normal office periods, there is no such thing as organisational data that provides information for the employees whether the meetings are cancelled or rescheduled in order for others to use unoccupied spaces that are marked reserved. It's pretty common for people to get lost in large complex buildings.

One way to resolve these issues is by implementing indoor positioning which can help to detect the personnel to reduce the risk of both rescuers and victims at the

time of evacuation, tracks real-time occupancy data to utilize unused office spaces, office equipment and save active electricity, and navigates to their respected destinations. A set of coordinates in a room is required to support for a room-level localization. BLE beacons are suitable for this type of problems rather than other indoor positioning sources such as WiFi and RFID for unavoidable disadvantage such as continuous power supply. An effective machine learning model is identified based on its location estimation, performance and computation time to overcome the problems of indoor positioning in an office building.

## 1.2 Aim and Objectives

The main aim of the thesis is to implement the indoor positioning system using BLE beacons in an Office floor plan and to evaluate the performance of the chosen machine learning models for location estimation using collected data.

**Objectives**

- To Perform literature study on appropriate machine learning models that are suitable for indoor positioning system using BLE beacons.
- To evaluate the performance of the chosen machine learning algorithms that delivers the location estimation by comparing with performance metrics such as Euclidean distance error, CDF curve, RMSE and MAE.

## 1.3 Research Questions

The research questions proposed for this thesis to achieve the aim of this thesis are as follows:

**RQ1:What are the suitable machine learning models used for indoor positioning with BLE beacons in an office environment?**

**Motivation:** The motivation for this research question is to do a systematic literature study on the appropriate machine learning models that are related to indoor positioning.

**RQ2: What are the performances of these chosen machine learning algorithms for location estimation in an office environment?**

**Motivation:** The motivation of this research question is to implement and evaluate the chosen machine learning models for indoor positioning using BLE beacons. The collected RSSI values and the (x,y) location coordinate values are applied to these machine learning models to predict the location of the receiver. The performance of these models are tested by the performance metrics such as Euclidean distance error, CDF curve, RMSE and MAE.

The research hypothesis is stated as follows:

- **Null Hypothesis:** The Machine learning models if applied predicts the location of the receiver with least distance error.
- **Alternate Hypothesis:**The Machine learning models if applied predicts the location of the receiver with huge distance error.

# Chapter 2

# Background

This section provides a brief groundwork to acquire the knowledge on the indoor positioning using BLE beacons with machine learning. However, before we begin to discuss the core concepts of this thesis, it is worthwhile to give a brief introduction to the industrial context considered in this study.

With the rapid improvement of indoor position estimation and with the proliferation of smart phones, the indoor location-based services have become desirable [34]. There are two reasons why the indoor positioning system is exciting. Most importantly, the technology enables a device to make location estimation. Even under perfect conditions, the GPS is often limited to outdoors. IPS can enable a distance estimation within centimetres. This technology would provide enormous benefits to potential indoor area applications and help guide navigation in Offices, museums, hospitals and malls [34].


Figure 2.1: Indoor Positioning System [1]

Indoor localization has various benefits in multiple sectors like commercial buildings, malls, hospitals, museums and airports. There are many indoor positioning systems such as Wi-Fi, BLE, RFID, wireless local area network (WLAN) and Ultra-Wideband (UWB) [34][35][36][37].

## 2.1 Characteristics for Indoor Positioning System

The characteristics that are important for indoor positioning are shown below:

- **Accuracy and Precision :** A good positioning system should be able to locate the user in the environment with high probability and precision [38] [39].
- **Calibration complexity:**The system should be mounted and configured as easily as possible. If the system takes lots of time and effort then, its calibration complexity is considered being not good [39].
- **Liveliness and response time :**The update rate must be quite high for the real-time tracking with at most a few seconds between each update [39].

## 2.2 Bluetooth Technology

Bluetooth is a technology that enables the communication of electronic devices without wire. Based on low cost transceiver microchips, it was designed for low power consumption. Bluetooth communicates frequencies between 2,402 GHz and 2,480 GHz using radio waves within the 2.4 GHz ISM frequency band, a frequency band set aside by international agreement for commercial, scientific and medical equipment [40].

It has been developed for continuous streaming data applications so that a large amount of data exchange over the close range can be carried out. Therefore, it is used by portable headphones, hands-free communication through your car, and wireless data transmission. Before setting up a network, any device in discoverable mode will then reply by returning information about itself, such as its name and address. Subsequently, the two devices can pair with each other, a process in which they create a common link key that is stored on each device [41].

Bluetooth is divided into three different classes, each having a different range the surrounding environment might be affected by this range, as the signals are susceptible to propagation effects. Even though class 3 devices would be suitable for indoor positioning purposes due to the small range, these devices are very rare, and the vast majority of available devices are of class 2 [40].

| Class | Range |
|---|---|
| Class 1 | 100 m |
| Class 2 | 10 m |
| Class 3 | 5 m |

Figure 2.2: Bluetooth range

## 2.3 iBeacons

iBeacon is Apple's brand name for micro-location-based technology and mobile device communication in the physical world. This software can be considered as the next stage of QR code technology or NFC technology development. iBeacon is using the Bluetooth Low Energy standard, a new Bluetooth 4.0 version. The way the peripheral device signals its existence to the other devices is also the opposite of how it is in the Bluetooth Classic original [42].

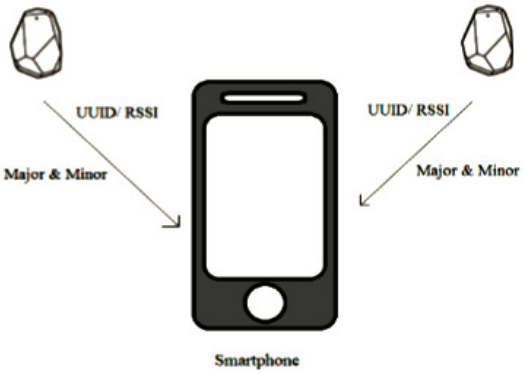


Figure 2.3: iBeacons for Indoor Positioning

iBeacon is a small device that transmits data in a given radius and at regular intervals. As soon as a smartphone gets into this area, it can obtain these data and can perform an action based on the information [42]. Using coin cell batteries, iBeacon technology can be operated for a month or longer, or function for months at a time using larger batteries or powered externally for extended periods. The iBeacon identifying information is provided by the UUID, major and minor values. The following information is provided by an iBeacon via Bluetooth Low Energy [43]:

| Field | Size | Description |
|---|---|---|
| **UUID** | 16 bytes | Application developers should define a UUID specific to their app and deployment use case. |
| **Major** | 2 bytes | Further specifies a specific iBeacon and use case. For example, this could define a sub-region within a larger region defined by the UUID. |
| **Minor** | 2 bytes | Allows further subdivision of region or use case, specified by the application developer. |

Figure 2.4: iBeacon Content Description

## 2.4 Android Platform

Android is a Google-developed mobile operating system, mainly for smartphone and tablet touchscreen devices. The system was originally created by Android Inc.

Google bought it in 2005, before the release of the mobile platform in 2007. A collection of core libraries and functionalities can be accessed, expanded, and customized to develop applications for modern mobile devices via Android API (Application Programming Interface). Applications ("apps"), which expand a device's functionality, are written using the Android SDK, mostly accompanied by the language of Java programming [22].

Due to the wide range of devices and sensors available on the market, ease of use and widespread use of the platform, I decided to use the Android platform in this thesis. In this study, due to lack of time, I used the Google Play Store Android application, whereas a proper approach might have been to create an application that allowed users to obtain beacon information such as RSSI value, UUID, distance and to save it for further post processing in a cloud server.

## 2.5 Beacon Locator

Beacon Locator is a beacon scanning, tracking and management application for Android [2]. This application was implemented using Mvvv pattern and data binding. The application scans and locates beacons and provides detailed information on the characteristic beacons. In addition, the application also gives the distance between the beacon and smartphone in which the app is installed [44].

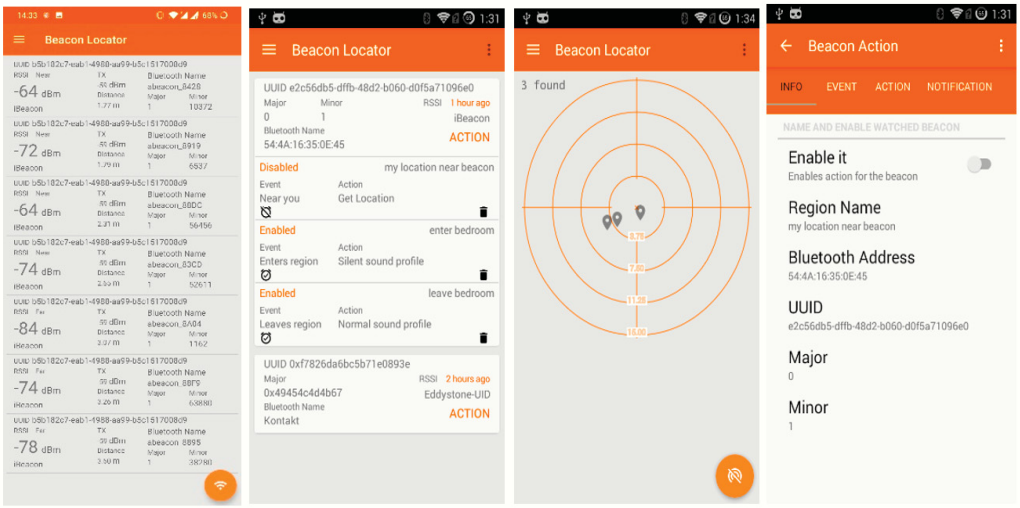


Figure 2.5: Beacon Locator Application [2]

## 2.6 Indoor Positioning Methods

### 2.6.1 Fingerprinting

The fingerprinting method is based on a radio mapping, which is a collection of fingerprints. A fingerprint is a collection of radio signals collected at a specific location in which each signal is correlated with the device it was emitted from. Fingerprints

can be matched to evaluate their resemblance, since identical fingerprints are supposed to come from nearby locations [45]. This method involves two phases, which is an offline phase, in which the radio map is generated and an online phase, in which the actual estimation of the location takes place [40].

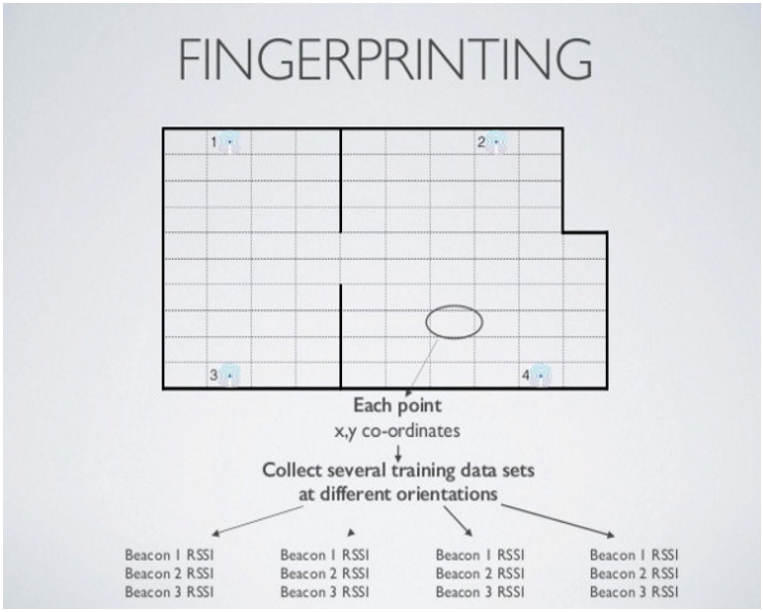


Figure 2.6: Fingerprinting [3]

The following sections provides a more detailed description of the two phases.

- **Offline Phase :** We deployed n BLE beacons in a single-floor area implementing a positioning system by providing full signal coverage of the area. The floor plan area is divided into small grid restricting possible positions to the grid's intersection points. At each point, m fingerprints are created and are stored in a database that is used as a radio map, and associate each of them with the point at which they were created. Every point on the radio map is identified with (x, y) coordinates [40][46]. There are instances where fingerprints are not created either by default or because areas covered by furniture or other barriers are not possible. Moreover, sufficient fingerprints must be created at one point to have a statistically valid sample of measurements [45].

- **Online Phase:** The online phase is where the estimation of the position takes place. The fingerprints are represented as vectors that contain the RSSI values obtained from i beacons for each element i. A value of zero is assigned to the elements corresponding to beacons not visible from the given location. The smartphone uses the beacon locator, which is an android application to estimate the location and to obtain the RSSI of all available beacons [40].

### 2.6.2 Triangulation

Triangulation method calculates the location based on the distance between specific reference points, where these distances yield an intersecting point where the receiver

is located. Signal strength-based probability methods combine signal strength measurements to calculate the probability of the user's position and determine the most likely location. This method form circles centred at the access points, where the radius of each circle is determined by the measured signal strength of the receiver or the time elapsed transmitting the signal between the access point and the receiver. An intersection point arises when there are three or more access points within a certain range and the intersection point gives the estimated location of the receiver. In real-world scenarios, it is almost impossible to obtain a single interaction point due to errors in measurements. The signal strength measurement can be affected by obstacles[4].

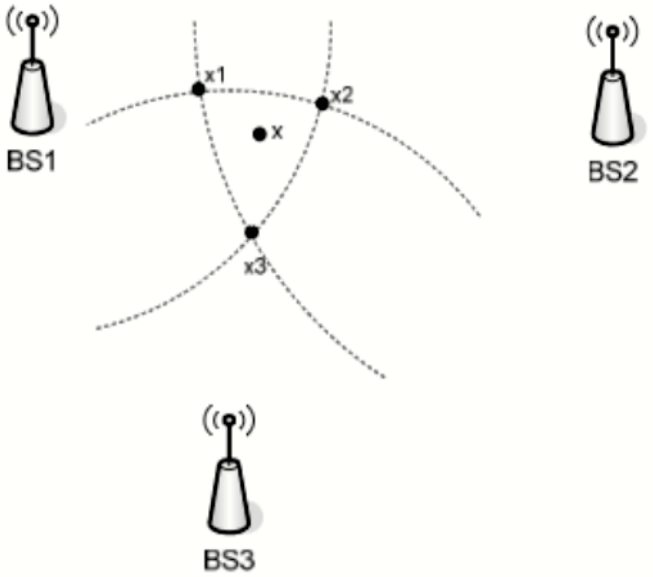


Figure 2.7: Triangulation Method [4]

### 2.6.3 Trilateration

Trilateration is the determination of absolute or relative locations by measuring the distances, using geometry. In this method, three fixed points are needed to determine an indoor position. The main idea behind this method is to calculate distances between these access points (AP) and the receiver to provide an area of localization. This distances can be provided by RSS, time of arrival of radio signals from transmitters (ToA) or time difference of arrival of several radio signals (TDoA). By using this method, one considers three or more APs allocated in the building. The signal strengths of these points are decreasing exponentially depends on the distance between the transmitter, receiver, and random noise factor. The distance estimated by signal strength is presented as a circle with a radius around the AP. The intersection of three AP radius's provides a point or an area of the receiver [5].

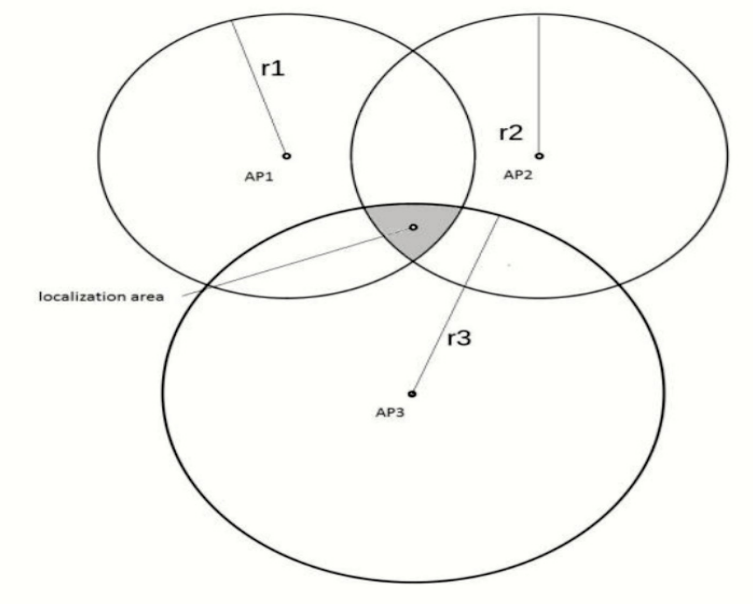


Figure 2.8: Trilateration Method [5]

## 2.7 RSSI

RSSI stands for Received Signal Strength Indicator. As seen on the receiving device, e.g. a smartphone, it is the strength of the beacon signal. The frequency of the signal depends on the value of distance and transmitting power. The RSSI ranges from -26 (a few inches) to -100 (40-50 m) at maximum broadcasting power (+ 4 dBm). Using another value defined by the iBeacon standard called Measured Power, RSSI is used to approximate distance between device and beacon [47]. The further the device is from the beacon, the more unstable is the RSSI.

## 2.8 Machine Learning

"Machine learning, in artificial intelligence, discipline concerned with implementing computer software that can learn autonomously," according to Britannica Academic [48][49]. Therefore, by observing a training set of examples, a machine first learns to perform a task with data it hasn't encountered before in clear terms [50].

- **Supervised Learning:** In supervised learning, the aim is to infer function from labelled training data. The training data consists of the labels input vector X and output vector Y. The vector Y label is an illustration of their respective input example from input vector X and together with the input vector it forms a training example. The goal is to learn a general rule, which maps inputs to their respective outputs [51].

- **Unsupervised Learning:**Unsupervised learning is the training of a machine learning algorithm using information that is not labelled and allowing the algorithm to act on that information without guidance [52]. In simple words, the goal of unsupervised learning is to find patterns in the data without any labels.

Considering a dierent perspective of machine learning, the problems of machine learning can be categorized into two categories:

- **Classification**: Classification predictive modelling is the task of approximating a mapping function (f) from input variables (X) to discrete output variables (y) [53]. For example, spam ltering i.e. classifying a mail into spam and not spam.
- **Regression**: Regression predictive modelling is the task of approximating a mapping function (f) from input variables (X) to a continuous output variable (y) [53]. For example, predicting the cost of the house in the next years.

## 2.9 Machine learning Models

### 2.9.1 Multilayer Perceptron

An MLP network is a deep, artificial neural network. It is composed of more than one perceptron. They are composed of an input layer to receive the signal, an output layer that makes a decision or prediction about the input, and in between those two, an arbitrary number of hidden layers that are the true computational engine of the MLP [6]. MLP's are often applied to supervised learning problems: they train on a set of input-output pairs and learn to model the correlation (or dependencies) between those inputs and outputs [6]. In MLP based localization, an input vector of the RSSI measurements is multiplied with the input weights and added into an input layer bias, provided that bias is selected. The obtained result is then put into hidden layer's transfer function. The product of the transfer function output and the trained hidden layer weights is added to the hidden layer bias (if bias is chosen). The obtained output is the estimated user location.

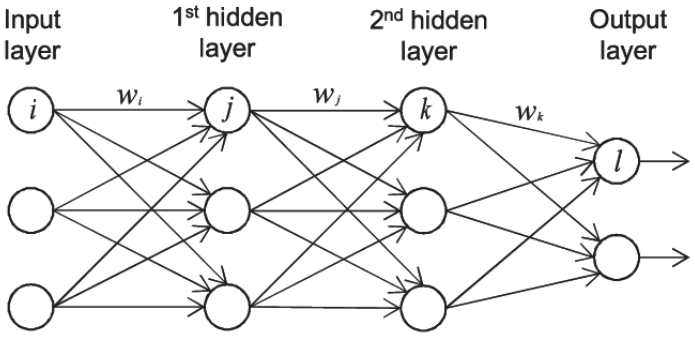


Figure 2.9: Multilayer Perceptron [6]

### 2.9.2 Long Short-Term Memory

The LSTM network is a type of recurrent neural network used in deep learning because very large architectures can be successfully trained [54]. LSTMs are explicitly designed to avoid the long-term dependency problem. Remembering information

for long periods of time is practically their default behaviour, not something they struggle to learn [52]. An LSTM module (or cell) has 5 essential components which allows it to model both long-term and short-term data [54].

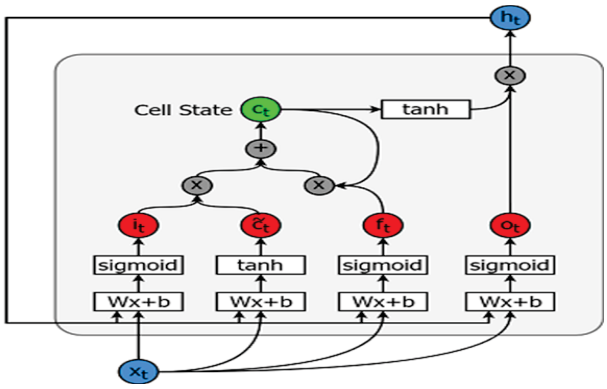


Figure 2.10: LSTM Module [7]

- **Cell state (ct)** : This represents the internal memory of the cell which stores both short term memory and long-term memories [55].
- **Hidden state (ht)** :This is output state information calculated w.r.t. current input, previous hidden state and current cell input. Additionally, the hidden state can decide to only retrieve the short or long-term or both types of memory stored in the cell state to make the next prediction [55].
- **Input gate (it)** :Decides how much information from current input flows to the cell state [55].
- **Forget gate (ft)** :Decides how much information from the current input and the previous cell state flows into the current cell state [55].
- **Output gate (ot)** :Decides how much information from the current cell state flows into the hidden state, so that if needed LSTM can only pick the long-term memories, or short-term memories and long-term memories [55].

### 2.9.3 Gradient Boosting

Gradient boosting is a machine learning technique for regression and classification problems, which produces a prediction model in the form of an ensemble of weak prediction models, typically decision trees [56]. The Gradient Boosted Regression is one of the most effective machine learning models for predictive analytics, making it an industrial workhorse for machine learning. Unlike Random Forest which constructs all the base classifier independently, each using a subsample of data, Gradient Boosted Regression Trees (GBRT) uses a particular model ensembling technique called gradient boosting [57].

### 2.9.4 Ada Boosting

An AdaBoost regressor is a meta-estimator that begins by fitting a regressor on the original dataset and then fits additional copies of the regressor on the same dataset but where the weights of instances are adjusted according to the error of the current prediction. As such, subsequent regressors focus more on difficult cases [58].

The core principle of AdaBoost is to fit a sequence of weak learners (i.e., models that are only slightly better than random guessing, such as small decision trees) on repeatedly modified versions of the data. The predictions from all of them are then combined through a weighted majority vote to produce the final prediction. The data modifications at each so-called boosting iteration consist of applying weights $w_1, w_2, \ldots, w_N$ to each of the training samples. Initially, those weights are all set to $w_i = 1/N$ , so that the first step simply trains a weak learner on the original data [59].

### 2.9.5 XG Boosting

XGBoost stands for "Extreme Gradient Boosting" and it is an implementation of gradient boosting machines. The XGBoost is a popular supervised machine learning model with characteristics like fast in computation, parallelization, and better performance [60]. XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It implements machine learning algorithms under the Gradient Boosting framework. XGBoost provides a parallel tree boosting (also known as Gradient Boosting Decision Tree) that solve many data science problems in a fast and accurate way [61].

# Chapter 3

# Related Work

A literature review was performed in this thesis to identify suitable machine learning model for indoor positioning with BLE beacons performed in an Office environment using machine learning algorithms:

Sahar, Ayesha, and Dongsoo Han [8],have focused mainly on deep and recurrent approaches for the improvement of the accuracy of positioning systems such as Long Short-Term Memory (LSTM) networks and other state of the art approaches such as K-Nearest Neighbor (KNN), probabilistic method, fuzzy logic, neural network and multilayer perceptron. A simple vanilla LSTM architecture is also compared with a stacked LSTM architecture on a Walking Survey dataset, where a person collects the Wi-Fi fingerprint while walking along the path. The dataset has a collection of RSS values at a particular location, radio mapping points of Wi-Fi fingerprints and location at an area or a building. They experimented with state-of-the-art machine learning approaches and concluded that LSTM works effectively in indoor positioning by a considerable margin than other machine learning approaches.

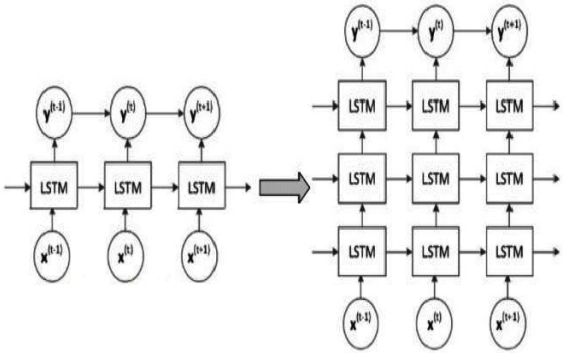


Figure 3.1: Simple LSTM vs Stacked LSTM [8]

Li, Dan, Le Wang, and Shiqi Wu [62], choose to locate the floor level of a mobile device using Wi-Fi fingerprint via machine learning methods, and explore the data size, feature dimension, model combination and parameter selection to maintain, if not improve, prediction accuracy, for different test environment. The purpose of this paper is that the author gave insights in IPS which aims at locating objects inside buildings wirelessly and have huge benefit for indoor location-aware mobile application which helps explore the immature system design. They used UJIndoorLoc

dataset and Principal Component Analysis (PCA) for feature selection, and build prediction models based on decision tree, gradient boosting, KNN and Support Vector Machine (SVM). From the experiment results, it indicates that combination of kNN and Gradient Boosting provides high prediction accuracy for Indoor Positioning. Whereas Gradient Boosting could highly increase the prediction accuracy and it also has small cross validation error for small data volume and is robust to missing data.

Taok, Anthony, Nahi Kandil, and Sofiene Affes [63], discusses the use of neural networks in an underground radio-localization system in a highly aggressive environment such as mines. They used UWB as the physical wireless propagation medium combined with fingerprinting-geolocation and neural networks in order to overcome the problems encountered in indoor environments. They conducted an experimentation by comparing MLP network and General Regression Neural Network(GRNN) - Radial Basis Functions (RBF) while considering both LOS and NLOS conditions. The results show that MLP network performs better accuracy than the other in LOS scenario whereas in NLOS scenario GRNN network produces higher accuracy.The results shows that the authors got an accuracy of 0.4m with a precision of 73% and the maximum error is 1.5m with 8% of recurrence on training data. On the other hand, results obtained by using the testing data have a value of 0.65m error for 70% of the time. The overall percentiles for error, where the authors report an error of less than 2 meters for 80% of the cases.

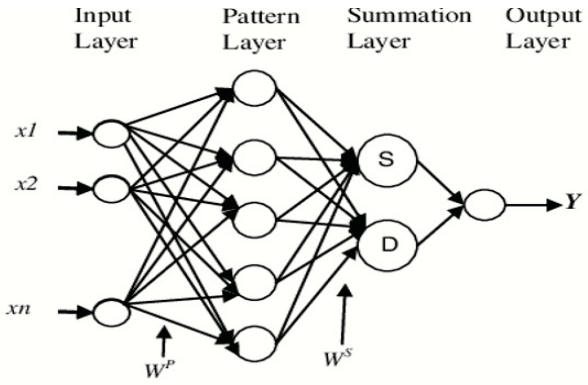


Figure 3.2: GRNN Architecture [9]

Feng, Yu, et al [10], have proposed an indoor localization technique based on improved AdaBoost algorithm. Since, the accuracy of AdaBoost algorithm depends on the weak hypothesis form all the weak learning [64], if there is noise in the fingerprint map, the performance of AdaBoost will decline. This noise can be avoided by the variability of indoor environment. Canovas, Oscar, Pedro E. Lopez-de-Teruel, and Alberto Ruiz [64],proposes to make use of RSSIs obtained during the training phase to build the resulting strong binary classifier.

From Figure 3.3, the authors describe that every dot represents reference point, and all triangles represents nearest neigbor points. After finding out all nearest neighbor points, the improved AdaBoost algorithm is used to separate out the optimal nearest

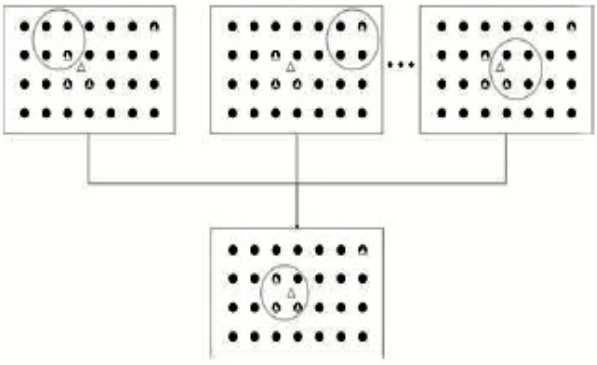

Figure 3.3: Improved AdaBoost Algorithm [10]

neighbour points without unfocused points. The author used this approach for the improved AdaBoost to ignore the individual unfocused points and develop the localization accuracy. An experiment is conducted with improved Adaboost algorithm in a Wi-Fi environment with RSS values and fingerprints. The experimental results show that the proposed algorithm achieves high localization accuracy than before. They propose to improve this algorithm in the future with more datapoints and more complex building rooms and floors.The improved AdaBoost algorithm is the most accurate of the five other models with localization error within 2 meters, where KNN and AdaBoost have the similar accuracy, the error of WKNN is bit smaller than KNN and traditional AdaBoost, and NN is the worst accurate of all in this experiment [10].

Faragher, Ramsey, and Robert Harle [26], provide a detailed study on comparison between BLE beacons positioning system, beacon density, transmit power and transmit frequency, and Wi-Fi fingerprinting method. The results show advantages to the use of BLE beacons for positioning over Wi-Fi network in the same indoor area. Li, Dan, Le Wang, and Shiqi Wu [62], describe that Wi-Fi positioning system (WPS) accuracy depends on the number of positions that have been entered into the database. The possible signal fluctuations that may occur can increase errors and inaccuracies in the path of the user. In [26], they conducted an experiment with BLE beacons deployment and Wi-Fi network, the results indicates that the BLE beacons shows significant improvement over Wi-Fi network and increase in number of beacons decrease positioning error.

Luckner, Marcin, Bartosz Topolski, and Magdalena Mazurek [65], propose to apply XGBoost algorithm in order to solve the issues such as classification and regression challenges for indoor positioning system. XGBoost algorithm was compared with random forrest and KNN algorithms using fingerprinting method with 2 datasets, one with observed access points and another with signals from academic network infrastructure. After testing with Random forrest and KNN, the results shows that XGboost obtained the best accuracy in the floor detection task. The XGBoost and Random forrest algorithms may be easier to use than KNN.

# Chapter 4

# Method

In this thesis, research questions were answered using the following approaches. First, a literature review will be used to study the current relevant literature in order to synthesize the results. Such findings of the literature review are used as a reference to interpret and test the second research method of the experiment.

## 4.1 Literature Review

At the beginning of the study, a literature review is undertaken to identify the different methods used to tackle Indoor positioning system. Different machine learning models were implemented exclusively for indoor positioning which distinguishes the advantages of each model and its concept of better execution for a particular issue. Machine learning models such as LSTM, MLP, Gradient boosting, XG boosting and Adaboosting are taken into consideration with Euclidean distance error, RMSE, MAE and CDF curve as performance metrics based on literature review. The articles for conducting a literature review were found by searching the following strings:

- "BLE beacon based indoor positioning using Machine Learning"
- "Methods to implement indoor positioning system using BLE beacons"
- "Neural networks for indoor positioning and navigation using BLE beacons"
- "Boosting Algorithms for indoor positioning and localization with beacons"

The articles were extracted from Digital Libraries such as BTH bibliotek, Google scholar, IEE Xplore, ResearchGate.

***Inclusion Criteria***

- Articles published in books, Journals, conferences and magazines.
- Articles which have been published between the years 2000 - 2019.
- Articles which are available in full text.
- Articles that are in English language.

***Exclusion Criteria***

- Articles without complete text.
- Articles not published in English.
- Articles excluding the 2000-2020 time frame.

## 4.2 Experiment

The main goal of this experiment is to evaluate the performance of machine learning models such as LSTM network, MLP network, Ada Boosting Regressor, XG Boosting Regressor and Gradient Boosting Regressor on the data of RSSI values and (X,Y) coordinates which are extracted from BLE beacons which are placed in an office floor plan of the NavAlarm Company. Results obtained from the experiment are analysed and compared to select the best-performed algorithm among them for the chosen data.The distance and RSSI values are dependent with each other but the RSSI values fluctuate so the best way is to treat them independently and coordinate them. The independent and dependent variables of the experiment are as follows:
**Independent Variables:**Distance, (X,Y) coordinates, RSSI values, Size of the experimental dataset, LSTM network, MLP network, Gradient Boosting, XG Boosting, Ada Boosting

**Dependent Variables:**Positioning error, Performance Metrics.

### 4.2.1 Dataset

The dataset used for this thesis was accumulated at the office space of Lead Start-up Incubator. A floor plan was created by measuring the length of the office space and used for the experimentation. The Walking survey dataset consists of seven BLE beacon signals (Beacon1- Beacon7) and the location co-ordinates (X,Y) of the receiver. The walking survey dataset, where a person collects the radio mapping fingerprints while walking along the path. The beacon signal measurements are RSSI and (X,Y) are the co-ordinates. The dataset was collected in the year 2019.

### 4.2.2 Data Preprocessing

The dataset is filtered in order to remove noise by using running average and if an RSSI value is missed in between, the lter inserts a minimum possible RSSI value in that vacant place. The dataset is also checked for null values which are also replaced with the minimum possible value or the above value in the data row.

### 4.2.3 Implementation

The experiment was conducted on the third floor of LEAD incubator at Mjärdevi Science Park, Linkoping. The devices provided by NavAlarm company are seven iBeacons, an android smartphone in order to conduct the experiment.

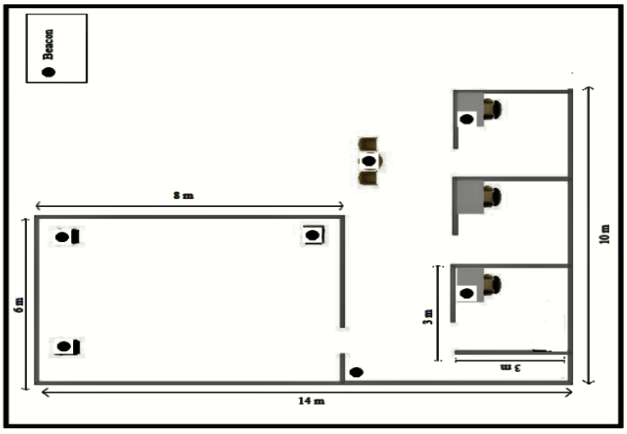


Figure 4.1: Floor Plan with BLE Beacons

### 4.2.4 Fingerprinting phase

In the fingerprinting phase the dataset collection is carried out by dividing the floor plan into many sub sections. Each such sub section is called a radio map. The dataset was collected by using walking survey method, where a person carrying the receiver (smartphone) walks on the certain path and record all the RSSI values from the nearest iBeacon devices along the way.

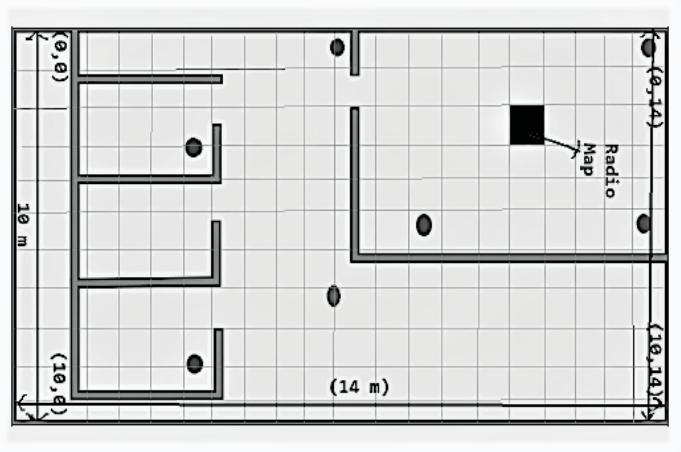


Figure 4.2: Fingerprinting

An android application on a smartphone device collects r samples of RSSI from k nearest beacon devices. Where map1 and map1rssi represents Radio mapping and D1,D2,...Dk represents Beacon devices. An n number of samples are collected at different points in each radio mapping section in the floor plan from k nearest BLE beacon devices. For example, the RSSI sample (map1rssi) collected at radio map (map1) shown below, where D1rssi1 represent the first RSSI value from device D1 [1].

$$map_1, map1_{rssi} = \sum_{i=1}^{m} |map_{ix}| \tag{4.1}$$

$$map_1, map1_{rssi} = \begin{Bmatrix} D1_{rssi1} D2_{rssi1} \dots Dk_{rssi1} \\ D1_{rssi2} D2_{rssi2} \dots Dk_{rssi2} \\ \dots \\ D1_{rssin} D2_{rssin} \dots Dk_{rssin} \end{Bmatrix}$$

Figure 4.3: Radio Mapping [11]

Thus, if there are m radio maps then we have m radio mapping pairs. Additionally, each radio map has (X, Y) coordinates and is represented as (mapix, mapiy). The radio mapping location coordinates (X,Y) and RSSI values from n beacon devices are used to train the selected machine learning models [1].

### 4.2.5 Machine Learning Phase

The collected dataset contains 1916 data points with (X,Y) location coordinates and RSSI values from seven iBeacons. For example, if we have total of m radio maps $map1, map2, \dots, mapm$ and given the r samples as input, the machine-learning algorithms return the probability of being at each radio map position.

$RSSIsample => Machine => p1, p2, \dots, pm$

where, pi represents the probability of being at the mapith radio mapping point, the estimated (X,Y) coordinates of the location can be predicted.

$$X = \sum_{i=1}^{m} |map_{ix}| \tag{4.2}$$

$$Y = \sum_{i=1}^{m} |map_{iy}| \tag{4.3}$$

## 4.3 Software Environment

### 4.3.1 Python

Python is a simple programming language aimed to be easy to read and implement. Since, it is an open source licensing, which means it is free to use for everyone [10]. Different features of this programming language are used for the experiments in this thesis.
The following libraries are used in the experiment in this thesis:

- **Pandas**:Pandas stands for "Python Data Analysis Library". Pandas is an open source library with BSD-licensed providing high-performance, easy-to-use data structures and data analysis tools with the Python programming language licence[66].

- **NumPy**: NumPy is the fundamental package for scientific computing with Python and is an open source library with BSD license. NumPy enables users to use efficient multi-dimensional container of generic data and arbitrary data-types. NumPy can seamlessly and speedily integrate with a wide variety of databases [67].
- **Matplotlib**:Matplotlib is a comprehensive library for creating static, animated and interactive visualizations in Python. It is a Python 2D plotting library which produces quality figures for publications across platforms in broad range of hardcopy formats and interactive environments [68].
- **Sklearn**:Sklearn is an open source library with BSD licence. It is simple to use and has efficient tools for predictive data analysis. Accessible to everyone and can be resued in various contexts [69].
- **XGBoost**: XGBoost is an optimized distributed gradient boosting library designed to be highly efficient, flexible and portable. It provides a parallel tree boosting that solve many data science problems in a fast and accurate way [70].

The machine learning models using Sklearn in this thesis are:

- Long Short-Term Memory
- Multilayer Perceptron
- Gradient Boosting
- Ada Boosting
- XG Boosting

The performances of these algorithms are calculated by using these following metrics imported by Sklearn:

- Mean Absolute Error
- Root Mean Square Error
- Euclidean Distance Error

### 4.3.2 Jupyter Notebook

The Jupyter Notebook is an open-source web application that lets users to create and share docs that have live code, equations, visualizations and narrative texts. It can be used for data cleaning and transformation, numerical simulation, statistical modeling, data visualization, machine learning, and much more [7]. In this thesis, Jupyter Notebook was used for writing the python code for the machine learning models.

## 4.4 Experimental Setup

- The experiment was performed by conducting K-fold cross validation with LSTM, MLP, Gradient boosting, XG boosting and Ada boosting models.
- All of these models were experimented with performance metrics and the results are compared for selecting the best algorithm for this dataset.

## 4.5 Performance Metrics

Performance metrics are used to evaluate different machine learning models. A regression model's efficiency can be understood by the error rate of the model predictions and the overall performance of the model. An outstanding regression model is the one where the discrepancy between the actual and the predicted values to train, validate and test data is small and neutral.

- **Mean Absolute Error:** MAE measures the average magnitude of the errors in a set of predictions without considering their direction. It's average of the absolute test sample difference between prediction and actual observation where all individual differences have equal weight [8].

$$MAE = \frac{1}{n}\sum_{i=1}^{n}|y_i - \hat{y}_i| \tag{4.4}$$

- **Root Mean Square Error:** RMSE is a quadratic scoring rule that also measures the average magnitude of the error same as MAE. It's the square root of the average of squared differences between prediction and actual observation [8].

$$RMSE = \sqrt{\frac{1}{n}\sum_{i=1}^{n}(y_i - \hat{y}_i)^2} \tag{4.5}$$

- **Euclidean Distance Error:** Euclidean distance is the straight line distance between two data points in a plane [9].

$$EuclideanDistance = \sqrt{(x_1 - x_2)^2 + (y_1 - y_2)^2} \tag{4.6}$$

## 4.6 Cross Validation

Cross-validation is a re-sampling procedure used to evaluate machine learning models on a limited data sample. It is a statistical method used to estimate the skill of machine learning models. Cross-validation is primarily used in applied machine learning to estimate the skill of a machine learning model on unseen data. That is, to use a limited sample in order to estimate how the model is expected to perform in general when used to make predictions on data not used during the training of the model[53].


Figure 4.4: K-Fold Cross Validation [12]

The procedure has a single parameter called k that refers to the number of groups that a given data sample is to be split into. As such, the procedure is often called k-fold cross-validation. When a specific value for k is chosen, it may be used in place of k in the reference to the model, such as k=10 becoming 10-fold cross-validation[53].The 5-fold cross-validation is used in the experimentation phase of this thesis.

## 4.7 Cumulative Distribution Function

The cumulative distribution function of a random variable is a method to describe the distribution of random variables. The advantage of the CDF is that it can be defined for any kind of random variable (discrete, continuous, and mixed). The CDF is the probability that the variable takes a value less than or equal to x. That is

$$F(x) = Pr[X \leq x] = \alpha$$

Note that the subscript X indicates that this is the CDF of the random variable X. For a continuous distribution, this can be expressed mathematically as

$$F(x) = \int_{-\infty}^{x} f(\mu) d\mu$$

For a discrete distribution, the CDF can be expressed as

$$F(x) = \sum_{i=0}^{x} f(i)$$

# Chapter 5

# Results

## 5.1 Long Short Term Memory

| | x | y | a | b | distance |
|---|---|---|---|---|---|
| **0** | 4.0 | 6.4 | 2.977802 | 9.089864 | 2.877543 |
| **1** | 3.8 | 6.4 | 2.765902 | 9.238783 | 3.021266 |
| **2** | 3.6 | 6.4 | 2.811192 | 9.054362 | 2.769090 |
| **3** | 3.4 | 6.4 | 3.000985 | 8.925412 | 2.556740 |
| **4** | 3.2 | 6.4 | 2.729645 | 9.252517 | 2.891035 |
| **5** | 3.0 | 6.4 | 2.907844 | 8.963206 | 2.564862 |
| **6** | 2.8 | 6.4 | 2.794900 | 8.997668 | 2.597673 |
| **7** | 2.6 | 6.4 | 3.050956 | 8.359767 | 2.010982 |
| **8** | 2.4 | 6.4 | 2.875990 | 8.906586 | 2.551380 |
| **9** | 2.2 | 6.4 | 2.734143 | 9.111733 | 2.763839 |

Figure 5.1: LSTM Predictions

LSTM neural network is trained with the dataset by using k-fold cross validation approach where 80% of the data was used for training and 20% of the data was used as the test set and the predictions are mentioned above. Euclidean Distance was used to evaluate the performance by calculating the Euclidean error using the Euclidean Distance formula mentioned in Section 4.8, where, (x, y) are the real coordinates and (a , b) are estimated coordinates of the location. LSTM neural network gave a Euclidean distance error of 3.59058.The CDF curve shows that the model is 100% accurate that of the distance error is within 7.5m.

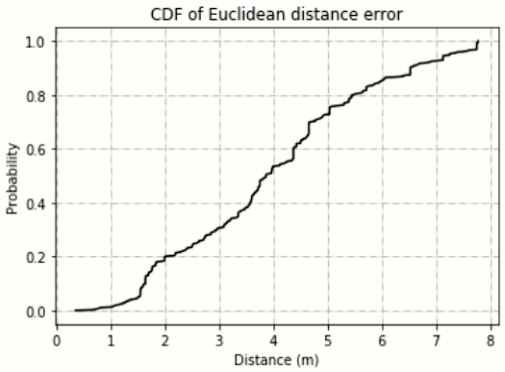


Figure 5.2: LSTM CDF curve

## 5.2 Multi Layer Perceptron

| | x | y | a | b | distance |
|---|---|---|---|---|---|
| **0** | 4.0 | 6.4 | 2.907690 | 5.721495 | 1.285889 |
| **1** | 3.8 | 6.4 | 3.148680 | 7.982374 | 1.711176 |
| **2** | 3.6 | 6.4 | 3.399438 | 6.104661 | 0.357002 |
| **3** | 3.4 | 6.4 | 3.403354 | 6.497801 | 0.097858 |
| **4** | 3.2 | 6.4 | 2.793560 | 9.953231 | 3.576401 |
| **5** | 3.0 | 6.4 | 3.579746 | 8.221983 | 1.911996 |
| **6** | 2.8 | 6.4 | 3.095003 | 8.644043 | 2.263351 |
| **7** | 2.6 | 6.4 | 2.770972 | 5.395809 | 1.018642 |
| **8** | 2.4 | 6.4 | 3.603565 | 6.799628 | 1.268176 |
| **9** | 2.2 | 6.4 | 4.495085 | 8.268764 | 2.959679 |

Figure 5.3: MLP Predictions

MLP neural network is trained with the dataset by using k-fold cross validation approach where 80% of the data was used for training and 20% of the data was used as the test set and the predictions are mentioned above. MLP neural network gave a Euclidean distance error of 4.05688.The CDF curve shows that the model is 100% accurate that of the distance error is within 11m.

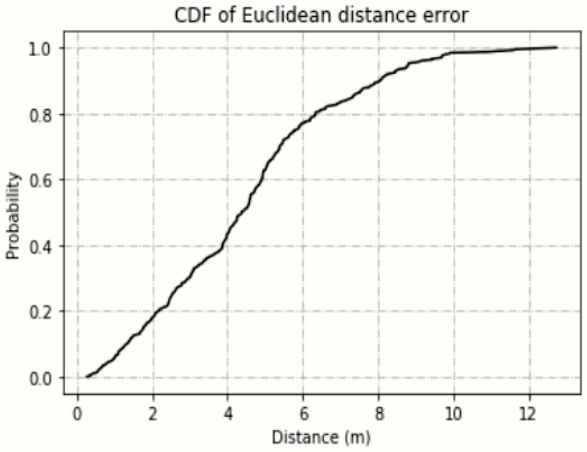


Figure 5.4: MLP CDF curve

## 5.3 Gradient Boosting Regression

| level_0 | index | x | y | a | b | distance |
|---|---|---|---|---|---|---|
| 0 | 1700 | 5.4 | 10.0 | 1.285726 | 7.831175 | 4.650919 |
| 1 | 1701 | 5.4 | 9.8 | 5.399999 | 10.200000 | 0.400000 |
| 2 | 1702 | 5.4 | 9.6 | 1.475050 | 10.073818 | 3.953446 |
| 3 | 1703 | 5.4 | 9.4 | 4.616060 | 10.431270 | 1.295407 |
| 4 | 1704 | 5.4 | 9.2 | 3.060063 | 10.327248 | 2.597304 |
| 5 | 1705 | 5.4 | 9.0 | 3.440859 | 8.771844 | 1.972381 |
| 6 | 1706 | 5.4 | 8.8 | 1.375764 | 9.132348 | 4.037936 |
| 7 | 1707 | 5.4 | 8.6 | 2.892573 | 6.814629 | 3.078107 |
| 8 | 1708 | 5.4 | 8.4 | 2.477897 | 6.779207 | 3.341505 |
| 9 | 1709 | 5.4 | 8.2 | 3.182561 | 6.409450 | 2.850106 |

Figure 5.5: Gradient Boosting Regression Predictions

Gradient Boosting Regression is trained with the dataset by using k-fold cross validation approach where 80% of the data was used for training and 20% of the data was used as the test set and the predictions are mentioned above. Gradient Boosting Regression gave a Euclidean distance error of 4.34932.The CDF curve shows that the model is 100% accurate that of the distance error is within 13m.

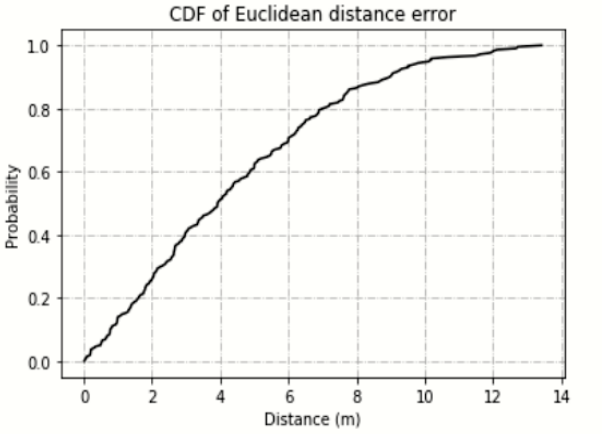


Figure 5.6: Gradient Boosting Regression CDF curve

## 5.4 XG Boosting Regression

| | index | x | y | a | b | distance |
|---|---|---|---|---|---|---|
| **0** | 1700 | 5.4 | 10.0 | -0.783223 | 11.335908 | 6.325891 |
| **1** | 1701 | 5.4 | 9.8 | 5.400006 | 10.199730 | 0.399730 |
| **2** | 1702 | 5.4 | 9.6 | 1.327939 | 11.737751 | 4.599094 |
| **3** | 1703 | 5.4 | 9.4 | 0.641101 | 7.302750 | 5.200537 |
| **4** | 1704 | 5.4 | 9.2 | 1.359909 | 4.803411 | 5.970958 |
| **5** | 1705 | 5.4 | 9.0 | 1.845396 | 4.980486 | 5.365790 |
| **6** | 1706 | 5.4 | 8.8 | 2.201900 | 10.050357 | 3.433837 |
| **7** | 1707 | 5.4 | 8.6 | 2.497127 | 8.505473 | 2.904412 |
| **8** | 1708 | 5.4 | 8.4 | 2.665275 | 11.544373 | 4.167229 |
| **9** | 1709 | 5.4 | 8.2 | 2.676593 | 9.421253 | 2.984695 |

Figure 5.7: XG Boosting Regression Predictions

XG Boosting Regression is trained with the dataset by using k-fold cross validation approach where 80% of the data was used for training and 20% of the data was used as the test set and the predictions are mentioned above. XG Boosting Regression gave a Euclidean distance error of 4.35923.The CDF curve shows that the model is 100% accurate that of the distance error is within 13m.

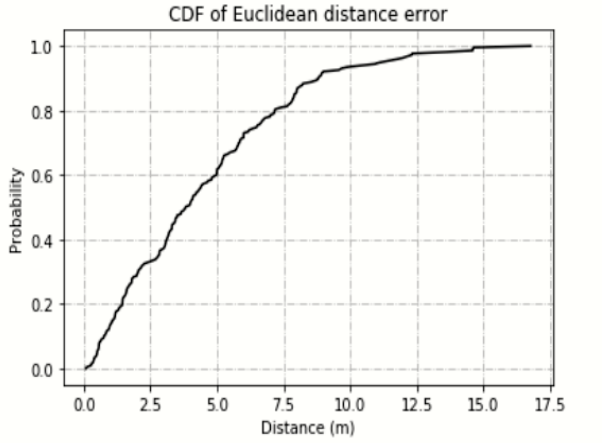


Figure 5.8: XG Boosting Regression CDF curve

## 5.5 ADA Boosting Regression

| | index | x | y | a | b | distance |
|---|---|---|---|---|---|---|
| **0** | 1700 | 5.4 | 10.0 | 2.706452 | 8.460649 | 3.102387 |
| **1** | 1701 | 5.4 | 9.8 | 2.845648 | 8.073070 | 3.083343 |
| **2** | 1702 | 5.4 | 9.6 | 2.706452 | 8.460649 | 2.924607 |
| **3** | 1703 | 5.4 | 9.4 | 3.107692 | 7.036155 | 3.292786 |
| **4** | 1704 | 5.4 | 9.2 | 2.706452 | 8.073070 | 2.919790 |
| **5** | 1705 | 5.4 | 9.0 | 2.706452 | 7.527507 | 3.069762 |
| **6** | 1706 | 5.4 | 8.8 | 3.107692 | 7.662500 | 2.559020 |
| **7** | 1707 | 5.4 | 8.6 | 3.107692 | 7.085164 | 2.747618 |
| **8** | 1708 | 5.4 | 8.4 | 2.706452 | 7.662500 | 2.792689 |
| **9** | 1709 | 5.4 | 8.2 | 2.706452 | 7.662500 | 2.746654 |

Figure 5.9: ADA Boosting Regression Predictions

Ada Boosting Regression is trained with the dataset by using k-fold cross validation approach where 80% of the data was used for training and 20% of the data was used as the test set and the predictions are mentioned above. Ada Boosting Regression gave a Euclidean distance error of 3.85725. The CDF curve shows that the model is 100% accurate that of the distance error is within 6.5m.

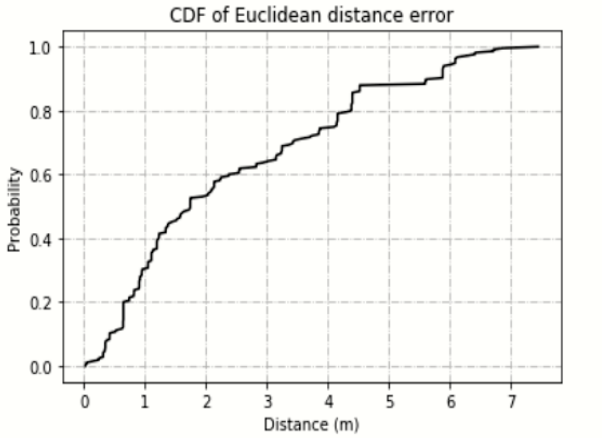


Figure 5.10: ADA Boosting Regression CDF curve

# Chapter 6

# Analysis and Discussion

## 6.1 Comparative study of Performance Metrics

| Algorithms | Root Mean Square Error | Mean Absolute Error | Euclidean Error |
|---|---|---|---|
| **Long Short-Term Memory** | 2.803 | 2.356 | 3.590 |
| **Multi Layer Perceptron** | 2.734 | 2.211 | 4.056 |
| **Gradient Boosting** | 3.706 | 2.731 | 4.349 |
| **XG Boosting** | 3.680 | 2.811 | 4.359 |
| **Ada Boosting** | 2.926 | 2.429 | 3.857 |

Table 6.1: Comparison of performance evaluation results

From **Table 6.1**, Neural networks performed better in the aspects of RMSE and MAE but when it came to Euclidean distance error, Ada boosting regression model performed well than other boosting models. The performance of Neural networks such as LSTM and MLP in overall performance metrics showed better than boosting regression models such as Gradient, XG, ADA boosting models. Ada boosting model has shown the minimum error in location estimation when compared to LSTM, MLP, Gradient boosting and XG boosting models. Gradient Boosting and XG boosting models showed poor performance in metrics such as RMSE, MAE and Euclidean distance error.

### 6.1.1 Performance analysis using Root Mean Square Error

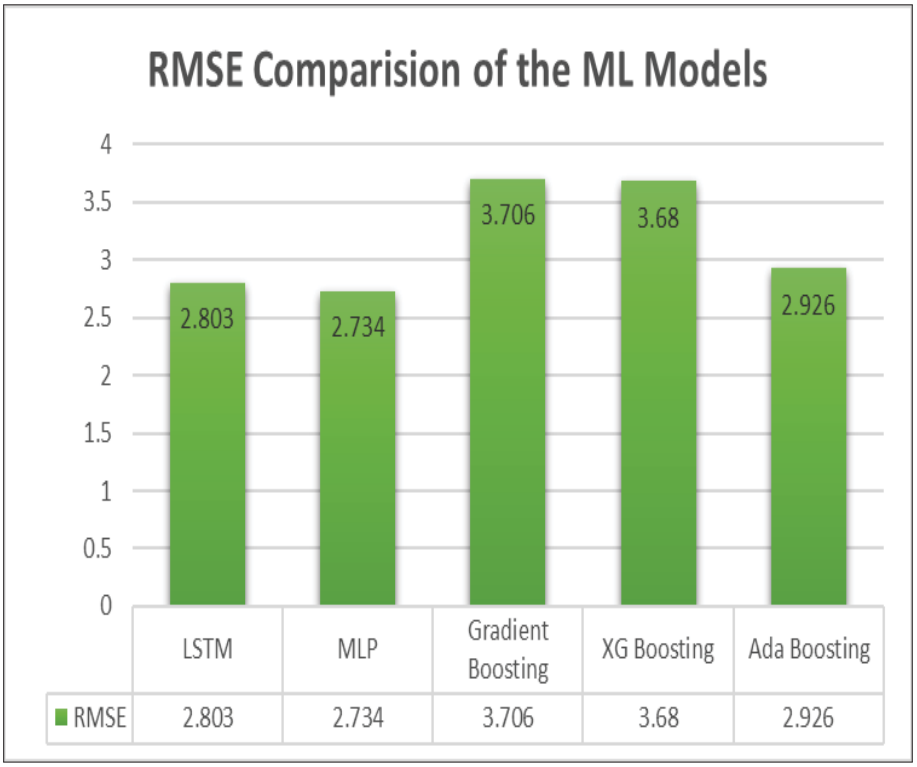


Figure 6.1: Comparison of RMSE obtained in different folds

The above **Figure 6.1** represents the RMSE from the results of the predictions produced by the models such as LSTM network, MLP network, Gradient boosting, XG boosting and Ada boosting on k-fold cross-validation tests. From the figure, it can be that MLP network has minimum RMSE when compared to other models. Both Gradient boosting and XG boosting has the highest RMSE with Gradient boosting being the worst performer.

### 6.1.2 Performance analysis using Mean Absolute Error

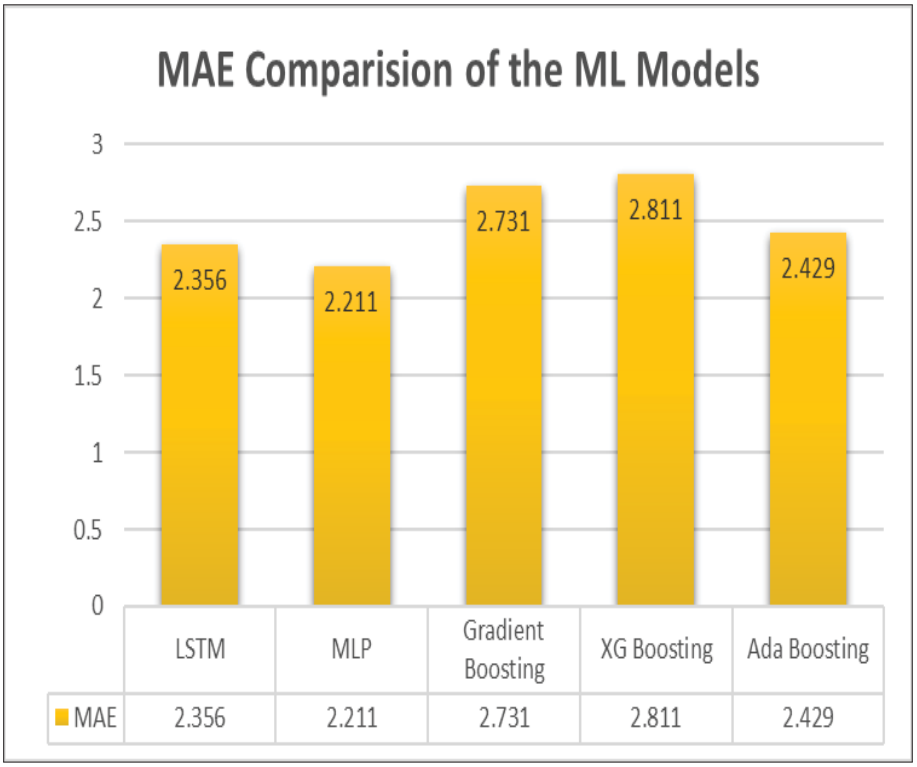


Figure 6.2: Comparison of MAE obtained in different folds

The above **Figure 6.2** represents the MAE from the results of the predictions produced by the LSTM network, MLP network, Gradient boosting, XG boosting and Ada boosting on k-fold cross-validation tests. It can be observed from the above figure that MLP network showed better performance with lesser MAE compared to other models. Whereas, both Gradient boosting and XG boosting models showing highest MAE with XG boosting being the worst performer.

## 6.2 Key Findings

In this thesis, MLP, LSTM and Ada Boosting showed lesser error in RMSE and MAE than the other models when compared. But when coming to Euclidean distance, LSTM has the least error i.e., 3.5m when compared with other models such as MLP, Gradient boosting and XG boosting and Adaboosting. From the results, it can be seen that the performance of LSTM is better because of its ability to capture long term dependencies as well as comparing of all models in terms of their convergence, precision, robustness, and general performance. Ada Boosting can be seen as good performer in terms of location prediction and accuracy when compared to all models based on CDF curve. Lastly, Gradient boosting and XG boosting regression models showed poor performance because of the overfitting problem and are harder to tune compared to other models.

## 6.3 Discussion

**RQ1: What are the suitable machine learning models used for indoor positioning with BLE beacons in an office environment?**

**Answer:** Based on the results obtained from the literature review five machine learning algorithms namely LSTM, MLP, Gradient boosting regressor, Ada boosting regressor and XG boosting regressor have been chosen for indoor positioning using BLE beacons in an office environment.

**RQ2: What are the performances of these chosen machine learning algorithms for location estimation in an office environment?**

**Answer:** From the results obtained, LSTM is chosen as the best suitable algorithm for location estimation in an office environment. In this experiment, MLP and LSTM has least error in RMSE an MAE when compared to Gradient boosting, XG boosting and Ada boosting models. The Euclidean distance error of LSTM model shows the least distance error among other models which is 3.5m and with 100 percentile of location error in CDF curve is within 7.5m. Followed by Adaboosting with 3.8m Euclidean distance error and with 100 percentile of location error in CDF curve is within 6.5m. Simultaneously results of other models with 100 percentile of location error in CDF curve are MLP with distance error of 4m and location error is within 11m error, Gradient boosting with distance error of 4.3m and location error is within 13m and XG boosting regressor has shown a worst performance with distance error being 4.3m and location error is within 14m. The least RMSE across the k-fold cross-validation is 2.734 and least MAE is 2.211 for MLP network model. The highest RMSE across the k-fold cross-validation is 3.706 for Gradient boosting, followed by 3.680 for XG boosting and the highest MAE being 2.731 and 2.811 for gradient boosting and XGboosting. The Gradient and XG boosting regression models when compared with the performance metrics has shown the worst performance. The performances of all the models have been discussed in **Section 6.1**.

**To be noted:**

- The experiment was conducted in a small space where is not a lot of movement so, this cannot be concluded that this can work in any office environment.
- The data-set used for this thesis is not good because it was collected manually with a walking survey method so the data cannot be trusted.
- From the related work, improved AdaBoost which was experimented by Feng, Yu, et al [10] has least error with 2m, when compared to results obtained from the experiment the least being LSTM with 3.5m distance error. Where the work performed earlier has better localization rate than this one, but this work has more potential to perform better with right and huge data instances.
- Taok, Anthony, Nahi Kandil, and Sofiene Affes [63], work shows that least error being 2m for 80% of the cases when compared to the our experimental results showing Adaboost with 3.8m distance error and with 100% accurate that the location error is within 6.5m which is relatively huge. But the work performed by them [63], they used UWB based indoor localization which is subjected to poor accuracy when people are obstacles are present. So, with lot of obstacles and movement present BLE beacon based indoor positioning system shows good performance.

## 6.4 Limitations

- A decent amount of data is required to effectively train and evaluate the machine learning models. The experiment conducted in this research had fewer data instances. This could have affected the results. In order to obtain good results in the experiment, it is required to have more data instances.
- Due to the unavailability of resources such as the prebuilt application that can track beacons and show real-time data, and a greater number of beacons than included in the experiment would benefit in obtaining better positioning accuracy.

## 6.5 Validity threats

Validity is an indicator of how well an assessment really tests what it should be measuring [71]

### 6.5.1 Internal validity

To overcome the threat of missing observations in the experiments, cloud backup is used which consists of all the logs copies of the experiment.

### 6.5.2 External validity

This validity is accomplished by using similar data to evaluate the algorithms and its performance. The threat of specifying all the dependent variables in this thesis in such a way that they are relevant in experimentation.

### 6.5.3 Conclusion validity

This problem may be posed if there is no appropriate selection of performance metrics that can lead to an understanding of the relationship between independent and dependent variables in the research. Multiple evaluation metrics have been used along with the proper structure of experimental setup and methodology, to avoid this threat.

# Chapter 7

# Conclusions and Future Work

In this research, LSTM network, MLP network, Gradient Boosting, XG boosting, and Ada boosting regression models are identified as appropriate algorithms to track office equipment in an office environment using a dataset containing RSSI values from BLE beacons that were used in experiments for indoor positioning. The location prediction is aimed to help detect the location of the people who are struck inside the building at the time of an emergency evacuation, to help people to get to their destinations without getting lost in large complex buildings and also to manage office equipment. Using performance metrics such as, Euclidean Distance error, RMSE and MAE for error estimation, the trained algorithms were assessed on 1916 data instances with seven beacon RSSI values and (x, y) location coordinates. After analysing the results, it is found that LSTM and MLP network showed the better results following by Ada Boosting regressor. Gradient boosting and XG boosting regression models being worst performers. The performance evaluation indicates that the Ada Boosting model has better precision for location prediction than LSTM, MLP, Gradient Boosting, XG Boosting. However, since this thesis is aimed to research a real-world issue, it can be concluded that LSTM is the algorithm of choice based of the overall performance and for efficiently predicting location in an office building.

**Future Work**
The future research can be done by gathering more data instances in large buildings to understand the environment by using deep learning where moving beacons can also be researched and implemented.

# References


[1] "Solutions." https://www.infsoft.com/solutions/basics/quick-start-indoor-positioning.

[2] "Beacon locator app." https://play.google.com/store/apps/details?id=com.samebits.beacon.locator\%5C&hl=en_US., 2020.

[3] "Indoor navigation with ble." https://developex.com/blog/indoor-navigation-with-ble/, journal=Developex blog, author=Admin, year=2017, month=Nov.

[4] Y. Wang, X. Yang, Y. Zhao, Y. Liu, and L. Cuthbert, "Bluetooth positioning using rssi and triangulation methods," in *2013 IEEE 10th Consumer Communications and Networking Conference (CCNC)*, pp. 837–842, IEEE, 2013.

[5] M. Shchekotov, "Indoor localization method based on wi-fi trilateration technique," in *Proceeding of the 16th conference of fruct association*, pp. 177–179, 2014.

[6] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[7] "Lstm explanation." https://doc.xuwenliang.com/docs/ai/2645, journal=AI - Xu Wenliang.

[8] A. Sahar and D. Han, "An lstm-based indoor positioning method using wi-fi signals," in *Proceedings of the 2nd International Conference on Vision, Image and Signal Processing*, pp. 1–5, 2018.

[9] I. Ladlani, L. Houichi, L. Djemili, S. Heddam, and K. Belouz, "Modeling daily reference evapotranspiration (et 0) in the north of algeria using generalized regression neural networks (grnn) and radial basis function neural networks (rbfnn): a comparative study," *Meteorology and Atmospheric Physics*, vol. 118, no. 3-4, pp. 163–178, 2012.

[10] Y. Feng, J. Minghua, L. Jing, Q. Xiao, H. Ming, P. Tao, and H. Xinrong, "Improved adaboost-based fingerprint algorithm for wifi indoor localization," in *2014 IEEE 7th joint international information technology and artificial intelligence conference*, pp. 16–19, IEEE, 2014.

[11] P. Sthapit, H.-S. Gang, and J.-Y. Pyun, "Bluetooth based indoor positioning using machine learning algorithms," in *2018 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pp. 206–212, IEEE, 2018.

[12] "introduction to support vector machines and kernel." `https://www.researchgate.net/publication/332370436_Introduction_to_Support_Vector_Machines_and_Kernel_Methods`.

[13] K.-W. Su, H.-Y. Hsieh, J.-C. Hsu, B.-H. Chen, C.-J. Chang, and J.-S. Leu, "Implementing an ibeacon indoor positioning system using ensemble learning algorithm," *Microsoft Indoor Localization Competition, Tech. Rep.*, 2017.

[14] L. Niu, "A survey of wireless indoor positioning technology for fire emergency routing," in *IOP Conference Series: Earth and Environmental Science*, vol. 18, p. 012127, IOP Publishing, 2014.

[15] C. Rizos, G. Roberts, J. Barnes, and N. Gambale, "Experimental results of locata: A high accuracy indoor positioning system," in *2010 International Conference on Indoor Positioning and Indoor Navigation*, pp. 1–7, IEEE, 2010.

[16] V. Renaudin, O. Yalak, P. Tomé, and B. Merminod, "Indoor navigation of emergency agents," *European Journal of Navigation*, vol. 5, no. ARTICLE, pp. 36–45, 2007.

[17] P. Baronti, P. Barsocchi, S. Chessa, F. Mavilia, and F. Palumbo, "Indoor bluetooth low energy dataset for localization, tracking, occupancy, and social interaction," *Sensors*, vol. 18, no. 12, p. 4462, 2018.

[18] K. Weekly, H. Zou, L. Xie, Q.-S. Jia, and A. M. Bayen, "Indoor occupant positioning system using active rfid deployment and particle filters," in *2014 IEEE International Conference on Distributed Computing in Sensor Systems*, pp. 35–42, IEEE, 2014.

[19] L.-H. Chen, E. H.-K. Wu, M.-H. Jin, and G.-H. Chen, "Intelligent fusion of wi-fi and inertial sensor-based positioning systems for indoor pedestrian navigation," *IEEE Sensors Journal*, vol. 14, no. 11, pp. 4034–4042, 2014.

[20] Y. Inoue, A. Sashima, and K. Kurumatani, "Indoor positioning system using beacon devices for practical pedestrian navigation on mobile phone," in *International conference on ubiquitous intelligence and computing*, pp. 251–265, Springer, 2009.

[21] S. Bozkurt, G. Elibol, S. Gunal, and U. Yayan, "A comparative study on machine learning algorithms for indoor positioning," in *2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA)*, pp. 1–8, IEEE, 2015.

[22] H. Espeland, "Navigation using bluetooth low energy beacons," Master's thesis, NTNU, 2018.

[23] Y. Wang, Q. Yang, G. Zhang, and P. Zhang, "Indoor positioning system using euclidean distance correction algorithm with bluetooth low energy beacon," in *2016 International Conference on Internet of Things and Applications (IOTA)*, pp. 243–247, IEEE, 2016.

[24] M. Choi, W.-K. Park, and I. Lee, "Smart office energy-saving service using bluetooth low energy beacons and smart plugs," in *2015 IEEE International Conference on Data Science and Data Intensive Systems*, pp. 247–251, IEEE, 2015.

[25] T. Gigl, G. J. Janssen, V. Dizdarevic, K. Witrisal, and Z. Irahhauten, "Analysis of a uwb indoor positioning system based on received signal strength," in *2007 4th Workshop on Positioning, Navigation and Communication*, pp. 97–101, IEEE, 2007.

[26] R. Faragher and R. Harle, "Location fingerprinting with bluetooth low energy beacons," *IEEE journal on Selected Areas in Communications*, vol. 33, no. 11, pp. 2418–2428, 2015.

[27] J.-H. Huh and K. Seo, "An indoor location-based control system using bluetooth beacons for iot systems," *Sensors*, vol. 17, no. 12, p. 2917, 2017.

[28] M. E. Rida, F. Liu, Y. Jadi, A. A. A. Algawhari, and A. Askourih, "Indoor location position based on bluetooth signal strength," in *2015 2nd International Conference on Information Science and Control Engineering*, pp. 769–773, IEEE, 2015.

[29] B. Benaissa, F. Hendrichovsky, K. Yishida, M. Koppen, and P. Sincak, "Phone application for indoor localization based on ble signal fingerprint," in *2018 9th IFIP International Conference on New Technologies, Mobility and Security (NTMS)*, pp. 1–5, IEEE, 2018.

[30] H. Torii, S. Ibi, and S. Sampei, "Indoor positioning and tracking by multipoint observations of ble beacon signal," in *2018 15th Workshop on Positioning, Navigation and Communications (WPNC)*, pp. 1–5, IEEE, 2018.

[31] F. Subhan, H. Hasbullah, A. Rozyyev, and S. T. Bakhsh, "Indoor positioning in bluetooth networks using fingerprinting and lateration approach," in *2011 International Conference on Information Science and Applications*, pp. 1–9, IEEE, 2011.

[32] Y. Zhang, L. Dong, L. Lai, and L. Hu, "Study of indoor positioning method based on combination of support vector regression and kalman filtering," *International Journal of Future Generation Communication and Networking*, vol. 9, no. 3, pp. 201–214, 2016.

[33] H. Mehmood, N. K. Tripathi, and T. Tipdecho, "Indoor positioning system using artificial neural network," *Journal of Computer science*, vol. 6, no. 10, p. 1219, 2010.

[34] A. A. Kalbandhe and S. C. Patil, "Indoor positioning system using bluetooth low energy," in *2016 International Conference on Computing, Analytics and Security Trends (CAST)*, pp. 451–455, IEEE, 2016.

[35] K. Al Nuaimi and H. Kamel, "A survey of indoor positioning systems and algorithms," in *2011 international conference on innovations in information technology*, pp. 185–190, IEEE, 2011.

[36] E. García, P. Poudereux, Á. Hernández, J. Ureña, and D. Gualda, "A robust uwb indoor positioning system for highly complex environments," in *2015 IEEE International Conference on Industrial Technology (ICIT)*, pp. 3386–3391, IEEE, 2015.

[37] C. Yang and H.-R. Shao, "Wifi-based indoor positioning," *IEEE Communications Magazine*, vol. 53, no. 3, pp. 150–157, 2015.

[38] "Accuracy vs precision." `http://www.diffen.com/difference/Accuracy_vs_Precision`.

[39] E. Dahlgren and H. Mahmood, "Evaluation of indoor positioning based on bluetooth smart technology," Master's thesis, 2014.

[40] A. Bekkelien, M. Deriaz, and S. Marchand-Maillet, "Bluetooth indoor positioning," *Master's thesis, University of Geneva*, 2012.

[41] B. Ray, "Bluetooth vs. bluetooth low energy: What's the difference?." `http://www.link-labs.com/blog/bluetooth-vs-bluetooth-low-energy`.

[42] P. Kriz, F. Maly, and T. Kozel, "Improving indoor localization using bluetooth low energy beacons," *Mobile Information Systems*, vol. 2016, 2016.

[43] A. Developer, "Getting started with ibeacon," *Retrieved May*, vol. 10, p. 2018, 2014.

[44] "Android application." `https://www.beaconzone.co.uk/blog/beacon-locator-android-app/`, 2020.

[45] T. van Dijk, "Indoor localization using ble," *Using Bluetooth Low Energy for Room-Level Localization*, 2016.

[46] M. Altini, D. Brunelli, E. Farella, and L. Benini, "Bluetooth indoor localization with multiple neural networks," in *IEEE 5th International Symposium on Wireless Pervasive Computing 2010*, pp. 295–300, IEEE, 2010.

[47] "How to calculate rssi." `https://iotandelectronics.wordpress.com/2016/10/07/how-to-calculate-distance-from-the-rssi-value-of-the-ble-beacon/`, 2020.

[48] "information technology." `https://www.itl.nist.gov/d`, 2020.

[49] "Machine learning." `https://www.britannica.com/technology/machine-learning`, 2020.

[50] https://wiki.python.org/moin/B, 2020.

[51] "iris waveformdata." https://ds.iris.edu/ds/nodes/dmc/data/types/waveform-data/, 2020.

[52] "Unsupervised learning." https://whatis.techtarget.com/definition/unsupervised-learning, 2020.

[53] J. Brownlee, "Difference between classification and regression in machine learning." https://machinelearningmastery.com/classification-versus-regression-in-machine-learning/, May 2019.

[54] X.-H. Le, H. V. Ho, G. Lee, and S. Jung, "Application of long short-term memory (lstm) neural network for flood forecasting," *Water*, vol. 11, no. 7, p. 1387, 2019.

[55] "(tutorial) lstm in python: Stock market predictions." https://www.datacamp.com/community/tutorials/lstm-python-stock-market, journal=DataCamp Community.

[56] "Gradient boosting." https://medium.com/mlreview/gradient-boosting-from-scratch-1e317ae4587d, journal=Medium, 2020.

[57] "boosted trees regression." https://turi.com/learn/userguide/supervised-learning/boosted_trees_regression.html, 2020.

[58] "Adaboost regressor." https://scikitlearn.org/stable/modules/generated/sklearn.ensemble.AdaBoostRegressor.html, 2020.

[59] "1.11. ensemble methods." https://scikit-learn.org/stable/modules/ensemble.html#adaboost.

[60] R. Python, "Regression example with xgbregressor in python." https://www.datatechnotes.com/2019/06/regression-example-with-xgbregressor-in.html, 2020.

[61] "Xgboost documentation." https://xgboost.readthedocs.io/en/latest/index.html.

[62] D. Li, L. Wang, and S. Wu, "Indoor positioning system using wifi fingerprint," *Stanford University, available online at*, 2013.

[63] A. Taok, N. Kandil, and S. Affes, "Neural networks for fingerprinting-based indoor localization using ultra-wideband.," *JCM*, vol. 4, no. 4, pp. 267–275, 2009.

[64] O. Canovas, P. E. Lopez-de Teruel, and A. Ruiz, "Detecting indoor/outdoor places using wifi signals and adaboost," *IEEE sensors journal*, vol. 17, no. 5, pp. 1443–1453, 2016.

[65] M. Luckner, B. Topolski, and M. Mazurek, "Application of xgboost algorithm in fingerprinting localisation task," in *IFIP International Conference on Computer Information Systems and Industrial Management*, pp. 661–671, Springer, 2017.

[66] "pandas." https://pandas.pydata.org/.

[67] "Numpy." http://www.numpy.org, 2020.

[68] "Visualization with python." https://matplotlib.org/.

[69] "Scikit learn." https://scikit-learn.org/stable/index.html.

[70] "Xgboost documentation." https://xgboost.readthedocs.io/en/latest/.

[71] I. N. Šerbec, M. Strnad, and J. Rugelj, "Assessment of wiki-supported collaborative learning in higher education," in *2010 9th International Conference on Information Technology Based Higher Education and Training (ITHET)*, pp. 79–85, IEEE, 2010.


Faculty of Computing, Blekinge Institute of Technology, 371 79 Karlskrona, Sweden