Master of Science in Computer Science
January 2020

Implementation of BLE Beacons in a Healthcare Centre for Indoor Localization using Machine Learning

K. Shiv Sidhartha

Dept. Computer Science & Engineering
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

This thesis is submitted to the Department of Computer Science & Engineering at Blekinge Institute of Technology in partial fulfillment of the requirements for the degree of Master of Science in Computer Science. The thesis is equivalent to 20 weeks of full-time studies.

Contact Information:
Author(s):
K. Shiv Sidhartha
E-mail: kasi17@student.bth.se

University advisor:
Prof. Yulia Sidorova,PhD
Dept. Computer Science & Engineering

Dept. Computer Science & Engineering
Blekinge Institute of Technology
SE–371 79 Karlskrona, Sweden

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

Abstract

Over the course of the last decade, Indoor based localization systems are becoming more prominent mainly due to the low cost and more effective devices such as BLE (Bluetooth Low Energy) beacons. The context of this thesis is to predict the location of patients and hospital equipment in a healthcare environment using machine learning algorithms. This can help to reduce the effort and the time needed by the staff which can be used to examine other patients. An experiment based on the fingerprinting approach using BLE beacons and perform data collection. Further, determining the positioning accuracy by training the dataset using state-of-the-art machine learning algorithms such as Random Forest, Ridge Regression, Linear Regression and K-nearest neighbor Regression. Finalising the results using performance metrics Euclidean distance error, Mean Absolute Error (MAE), Root Mean Squared Error (RMSE) and selecting the best model for this research. Fingerprinting approach for indoor localization is studied and applied for collecting the RSSI(Received Signal Strength Indicator) signals from the fixed beacons. A literature review is conducted and Random Forest Regression, Ridge Regression, Linear Regression and K- Nearest Neighbor regression models are selected and studied extensively. An experiment is performed to evaluate the performance of the machine learning algorithms. The results from the literature review show that regression models Random Forest Regression, Ridge Regression, Linear Regression and K- Nearest Neighbor Regression are suitable models and experimental results showed that Random Forest Regression performed better than Ridge Regression, Linear Regression and K- Nearest Neighbor Regression for predicting location of the beacons. Analyzing the results acquired and taking into account the real world scenario to which this thesis is intended, it can be stated that Random Forest Regression is the algorithm of choice for tracking hospital patients and equipment in a healthcare environment.

Keywords: machine learning, indoor positioning, fingerprinting.

Acknowledgments

Firstly, I would like to thank Sara M. Razavi for offering me a chance to supervise and conduct my thesis research and work at NavAlarm AB, Linköping. I would also like to thank her for sharing her experience, knowledge and guidance for the betterment of my thesis research. I would also like to express my deep sense of gratitude and thanks to Julia Sidorova for her supervision and encouragement at Blekinge Institute of Technology. Lastly, I would like to thank my family for their support and encouragement.

List of Figures

List of Tables

Contents

Chapter 1

Introduction

The demand for indoor positioning services have rapidly increased over the past decade as there are many fields that rely on positioning and localization. Some of the main examples of these location aware applications would be navigation of civilians inside buildings, navigation of customers inside malls and product localization in supermarkets [1] . Healthcare centres have many patients and hospital equipment that are always moving and their location is always changing. The staff of the hospital waste time trying to find them which can be used for attending other patients. Therefore, knowing the exact locations of the patients and equipment is necessary to increase the efficiency in a health centre [2].

Global Navigation Satellite Systems(GNSS) have proved to be effective and accurate positioning for an outdoor environment but the failure of these signals to enter buildings has made them fall short for indoor positioning. The most common consumer technology for indoor positioning are Wi-Fi, Radio Frequency Identification (RFID) and Bluetooth signals. Fingerprinting method is a well established and most commonly used technique for indoor positioning devices [3]. The WiFi technique is restricted with factors such as environment and power supply and the RFID technique is confined to the user with the requirement of special positioning equipment. Whereas, BLE (Bluetooth Low Energy) beacon technology has the ability to keep the signal power stable and easily set up a network which is crucial for indoor positioning [4].

A Bluetooth beacon runs on a coin battery which can operate up to two years depending upon the power usage and frequency. Basically, it is a low cost and low power technology [4]. BLE is a wireless technology that provides the location data to mobile devices. [5] It constantly sends out Bluetooth signals and each signal has RSSI (Received Signal Strength Indicator) and UUID (Universally Unique Identifier), etc [2]. Indoor positioning technique is classified into triangulation positioning and fingerprint positioning. The RSSI (Received Signal Strength Indicator) and distance relationship with transmitter and receiver is subjected to be changing due to external factors [6]. The precision of the triangular positioning reduces considerably as it depends heavily on the track model's rationality. Whereas, fingerprint positioning works on training phase (offline) and testing phase (online). During the training stage, RSSI vectors containing RSSI obtained on multi-point receivers are recorded before the testing stage as training data. The fingerprint positioning can be used in an indoor environment like a healthcare centre because the accuracy depends on the size and number of the RSSI vector and RSSI samples at different points [7].

In this thesis, fingerprinting technique is implemented using popular machine learning regression algorithms, Random Forest, Ridge Regression, Linear Regression and K- Nearest Neighbor. The machine learning algorithms are trained using the RSSI samples collected at various radio mapping points. After the training phase, the machine learning algorithms estimates the location of the beacons from the given RSSI samples as input. The system uses state-of-the-art machine-learning method for fingerprinting and location estimation. The experimental results show the working of the algorithms and their respective predicted locations. The dataset used in this thesis consists of RSSI samples and the locations from seven BLE beacons spread across the floor plan.

1.1 Problem domain

A Healthcare centre is subjected to a lot of movement of patients and hospital equipment from one place to another. The limitation of human resources, beds and equipment leads to constant change of location of the patients and resources. This wastes a lot of time of the medical staff which can be used for examining other patients. These issues can be solved using indoor localization and can help to pin point the exact location using positioning sources. Indoor positioning sources like Wi-Fi, Global Positioning System (GPS) and Radio Frequency Identification (RFID) have been subjected to unavoidable disadvantages like continuous power supply and user identification confinement. BLE beacons are the best alternative source since they consider RSSI and distance for effective indoor positioning, which can help to overcome these problems. Therefore, implementing machine learning algorithms for location estimation can help to overcome the problems of indoor positioning in a healthcare centre.

1.2 Aim and Objectives

The main aim of this thesis is to implement indoor positioning using BLE beacons on a floor plan and evaluate the performance of the chosen machine learning algorithms for location estimation using the collected data.

Objectives

- Perform literature study regarding the appropriate machine learning algorithms and BLE beacon technology used to address indoor positioning.
- Ultimately, evaluate the performance of the chosen machine learning algorithms by comparing the estimated and predicted positions for location estimation.

1.3 Research Questions

RQ1. What are the suitable machine learning techniques to address indoor positioning using BLE beacons?

Motivation: The motivation of this research question is to study the suitable machine learning models for indoor positioning.

RQ2. What are the performances of machine learning models for location estimation based on the collected data?

Motivation: The motivation for this research question is to implement and evaluate the selected machine learning models and apply them for indoor positioning using BLE beacons. The machine learning algorithms are applied to the collected RSSI(Received Signal Strength Indicator) samples along with ground truth to predict the location of the receiver, i.e the smartphone. The predicted values are tested for performance by applying the metrics Euclidean distance error, RMSE (Root Mean Squared Error), MAE (Mean Absolute Error) and plotting the CDF (Cumulative Distribution Function) graphs.

1.4 Thesis Outline

The thesis report consists of 7 chapters. The chapters are as follows. Chapter 1 is the introduction for the thesis and the motivation of the thesis is discussed briefly. Chapter 2 consists of the background and algorithms that were discussed in a more generalised manner. Chapter 3 showcases related work. Chapter 4 shows the research methods and methodology that is used in this thesis. Chapter 5 provides the results observed and the findings of this thesis. The results observed in Chapter 5 were analysed in Chapter 6. Chapter 6 also deals with validation of the results and Chapter 6 deals with discussions and limitations regarding this study and provides the answers to the research questions that motivated the research of this thesis. Finally, Chapter 7 concludes the thesis study and the future work that can be done is also discussed.

Chapter 2

Background

2.1 Indoor Positioning System

Indoor localization is the process of obtaining the location of the device or user in an indoor setting or environment. Over the past few decades, localization of indoor devices has been extensively investigated, mainly in industrial settings and for networks and robotics of wireless sensors [8]. Nevertheless, it was only less than a decade ago that the large-scale proliferation of smartphones and wearable devices with wireless communication capabilities made the location and tracking of such devices synonymous with the location and tracking of the relevant users and allowed a wide range of related applications and services.

Localization of users and devices has wide-ranging applications in the health sector, industry, disaster management, construction management, and a number of other sectors. Long-range Internet Of Things (IoT) technologies have not been developed with the provision of indoor positioning but short and medium range technologies, such as Bluetooth, Zigbee, WiFi, Ultra-Wideband (UWB), etc have been given the priority. Current short-range communication technologies can approximate the relative indoor position of an IoT unit in relation to certain reference points very accurately [8].

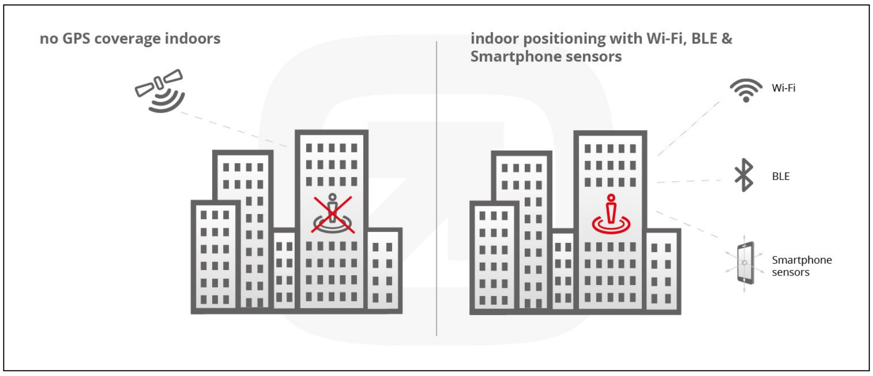


Figure 2.1: Indoor Positioning Systems
[9]

2.2 iBeacon

The iBeacon is BLE wireless technology introduced by Apple to create a different way for mobile devices to provide location-based information services. It acts as an ongoing transmitter of Bluetooth signals, each of which has a Universally Unique Identifier (UUID) and a Received Signal Strength Indicator (RSSI), etc. We used machine learning algorithms as our main position estimation tool in this research because it's easy to get the RSSI data from iBeacon [2].

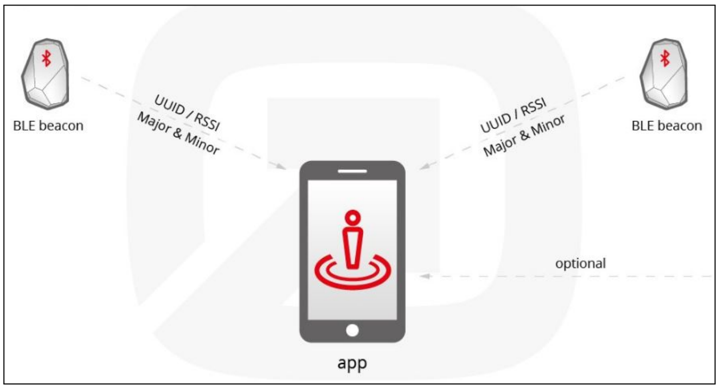


Figure 2.2: Indoor Positioning using Bluetooth Beacons
[9]

2.3 Indoor localization using BLE beacons

Indoor positioning using BLE beacons can be done by setting a few beacons inside a building. A smartphone application finds the beacons and gathers their RSSI readings. The distance between the receiver and every beacon is calculated. If the application has not received any information it uses a mobile device gyroscope and compass.This data is used to determine the direction of movement of a device. A map shows the location of both the beacons and the user, as well as the distance between them [10].

BLE beacons implemented for indoor localization has the following benefits:

- They can be in both client-based as well as server-based applications;
- Less effort to install;
- Supported by both Android and iOS applications;
- Energy efficient;

- Asymmetric peripheral design;
- Efficient to connect /discover;
- Low cost.

Use Cases for implementing BLE beacons :

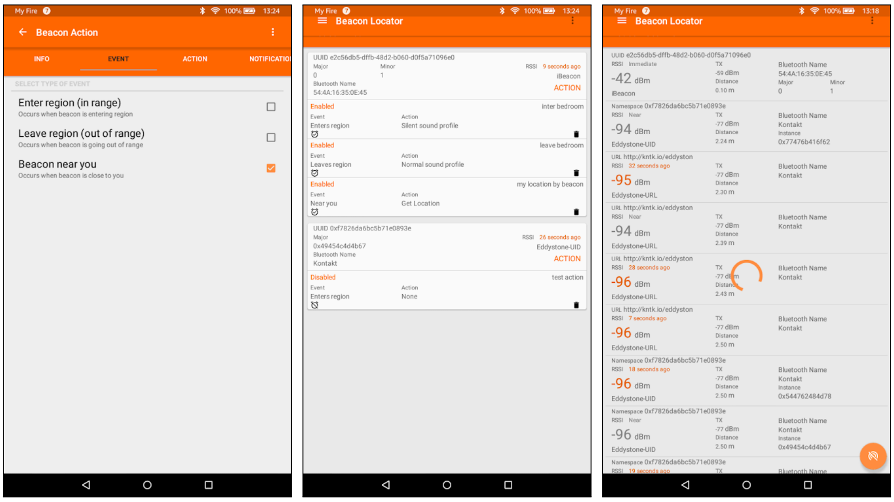
Healthcare and Medical

- Applications and hardware for patients and visitors.
- Tracking moving medical equipment.
- Integration into hospital single information system.

Other use cases that can use BLE beacons for positioning are Exhibitions and Conferences, Offices, Industries, Parking, Retail and Shopping etc.

2.4 Beacon Locator Application

Beacon Locator is an android based application which is used for scanning, tracking and managing of iBeacons. The application was developed by SameBits and will scan beacons (Eddystone, iBeacons or AltBeacons) and locate them. The application gives the information of distance between the beacon and smartphone in which this application is installed and being used [11].



SELECT TYPE OF EVENT

Enter region (in range) ☐
Occurs when beacon is entering region

Leave region (out of range) ☐
Occurs when beacon is going out of range

Beacon near you ☑
Occurs when beacon is close to you

UUID e2c56db5-dffb-48d2-b060-d0f5a71096e0
Major 0 Minor 1 RSSI 9 seconds ago
Bluetooth Name 54:4A:16:35:0E:45 iBeacon
ACTION

Enabled inter bedroom
Event Enters region Action Silent sound profile

Enabled leave bedroom
Event Leaves region Action Normal sound profile

Enabled my location by beacon
Event Near you Action Get Location

UUID 0xf7826da6bc5b71e0893e
Major 0x49454c4d4b67 RSSI 26 seconds ago
Bluetooth Name Kontakt Eddystone-UID
ACTION

Disabled test action
Event Enters region Action None

UUID e2c56db5-dffb-48d2-b060-d0f5a71096e0
RSSI Immediate -42 dBm iBeacon
TX -59 dBm Distance 0.10 m
Bluetooth Name 54:4A:16:35:0E:45 Major 0 Minor 1

Namespace 0xf7826da6bc5b71e0893e
RSSI Near -94 dBm Eddystone-UID
TX -77 dBm Distance 2.24 m
Bluetooth Name Kontakt Instance 0x77476b416f62

URL http://kntk.io/eddyston
RSSI 32 seconds ago -95 dBm Eddystone-URL
TX -77 dBm Distance 2.30 m
Bluetooth Name Kontakt

URL http://kntk.io/eddyston
RSSI Near -94 dBm Eddystone-URL
TX -77 dBm Distance 2.39 m
Bluetooth Name Kontakt

URL http://kntk.io/eddyston
RSSI 28 seconds ago -96 dBm Eddystone-URL
TX -77 dBm Distance 2.43 m
Bluetooth Name Kontakt

URL http://kntk.io/eddyston
RSSI 7 seconds ago -96 dBm Eddystone-URL
TX -77 dBm Distance 2.50 m
Bluetooth Name Kontakt

Namespace 0xf7826da6bc5b71e0893e
RSSI 18 seconds ago -96 dBm Eddystone-UID
TX -77 dBm Distance 2.50 m
Bluetooth Name Kontakt Instance 0x544762484d78

Namespace 0xf7826da6bc5b71e0893e
RSSI Near -96 dBm Eddystone-UID
TX -77 dBm Distance 2.50 m
Bluetooth Name Kontakt Instance 0x49454c4d4b67

Namespace 0xf7826da6bc5b71e0893e
RSSI 19 seconds ago TX Bluetooth Name

Figure 2.3: Beacon Locator Application

2.5 RSSI

RSSI is a received signal strength indicator (RSSI), generally a negative number ranging between 0 and 100 and can be used as an estimate of the distance separating the transmitter from the receiver (i.e. range) in localization systems. Other than dividing distance, RSSI is influenced by some other factors such as movement of people and objects in the midst of the environment's signals, temperature, and humidity [12].

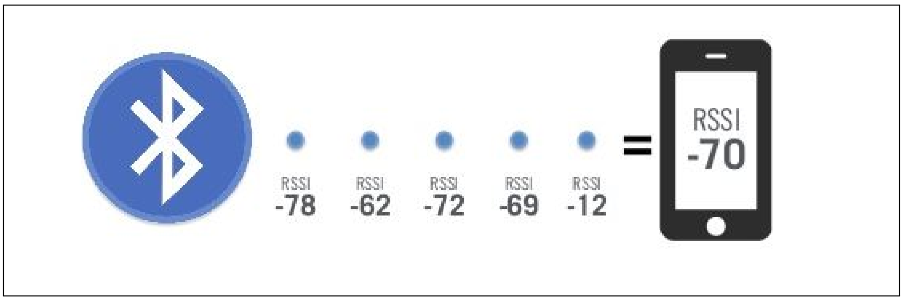


Figure 2.4: RSSI Signals from iBeacon recieved by Smartphone

2.6 Fingerprinting

The technique of fingerprinting is based on a radio map, a collection of fingerprints. A fingerprint is a series of radio signals at a specific location where each signal is associated with the device from which it was emitted. This method consists of an offline process in which the radio map is generated and an online phase in which the actual location is estimated [13]. Machine learning algorithms are implemented for position estimation based on the data collected using fingerprinting technique. Machine learning algorithms are initially trained at different radio mapping points with the RSSI samples. When the system is trained the machine estimates a mobile's position based on the input provided by the RSSI. The proposed system used in this thesis uses machine learning algorithms for the fingerprinting as well as location estimation [14].

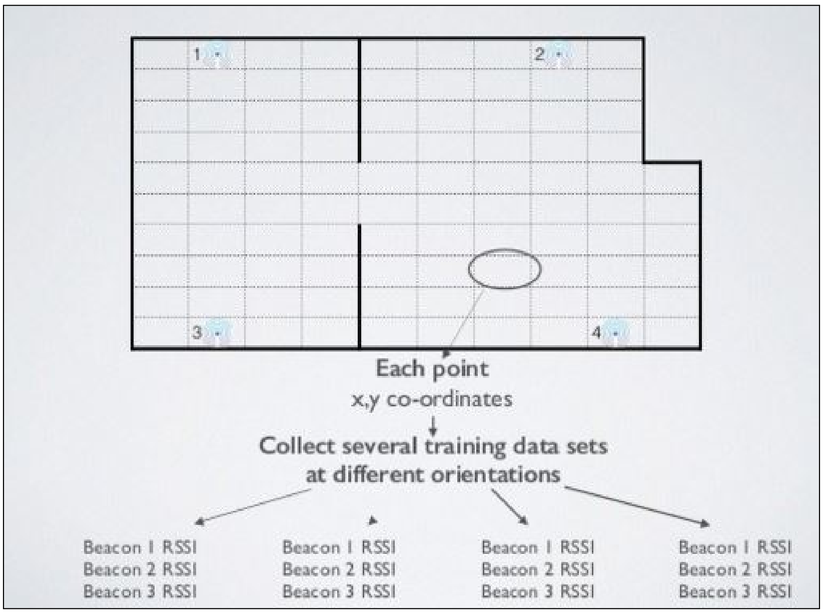


Figure 2.5: Fingerprinting method

2.7 Machine Learning

According to Britannica Academic, "Machine learning, in artificial intelligence (a subject within computer science), discipline concerned with the implementation of computer software that can learn autonomously" A supervised learning approach is when both input and output values are included in the results. By estimating the error between the predicted value of the model and the actual output value, the weights and biases of the model can be changed to minimize this error. [15]

Supervised learning: The goal in supervised learning is to infer from training data that is labeled, a function or mapping. The training data comprises of input vector X and output vector Y of labels or tags. A label or tag from vector Y is an indication of its respective input example from input vector X and forms a training example with the input vector. The algorithm's goal is to learn a general rule mapping inputs to their respective outputs [16].

Unsupervised learning: Unsupervised learning is training a machine learning algorithm using information that is neither marked nor labelled and that enables the algorithm to act without guidance on that information [17].

2.8 Machine Learning Models

2.8.1 Random Forest Regression

A Random Forest is an ensemble method that can perform both regression and classification tasks using multiple decision trees and a technique called Bootstrap Aggregation, usually referred to as bagging. Bagging involves training each decision tree on a different data sample in the Random Forest model where replacement sampling is performed.The random forest model is a type of additive model that predicts by combining decisions from a base model sequence [18].

Simply put, random forest creates multiple decision trees and merges them together to achieve a prediction that is more accurate and stable. Random Forest has almost the same hyper parameters as a decision tree or classifier for bagging. Random Forest brings randomness to the way the trees grow. It looks for the best feature among a random subset of features instead of looking for the most important feature when breaking a node. This leads to a wide variety that generally leads to a better model. Random forests are common for higher prediction speeds and low memory consumption because they only pick a subset of features from the entire node to break [19].

2.8.2 Ridge Regression

Ridge Regression is a technique for analyzing multiple regression data that suffer from multicollinearity. When multicollinearity occurs, least squares estimates are unbiased, but their variances are large so they may be far from the true value.By adding a degree of bias to the regression estimates, ridge regression reduces the standard errors. It is hoped that the net effect will be to give estimates that are more reliable.

$$min\Big(||Y - X(\theta)||_2^2 + \lambda||\theta||_2^2\Big) \tag{2.1}$$

Where X is a vector of weights, (θ) is the coefficient, (λ) is denoted by the alpha parameter in the ridge function. So, by altering, the values of the alpha penalty term are controlled. As the alpha value is higher, bigger is the penalty and accordingly coefficients magnitude is reduced. As the parameters shrink, it is mainly used for the of the multi-collinearity [20]. By doing the coefficient shrinkage model complexity is reduced and this process is called regularization.

2.8.3 Linear Regression

Linear regression is useful for finding relationship between two continuous variables. One variable is indicator or independent, and the other variable is response or dependent. This searches for a relationship that is predictive but not deterministic. It is said that the relationship between two variables is deterministic if the other can express one variable accurately. Use temperature in degree Celsius, for example, Fahrenheit can be measured accurately. In determining the relationship between two variables, the statistical relationship is not accurate. For instance, height-to-weight relationship. The core idea is to get a line that matches the data best. The best fit line is the one that has the smallest possible overall prediction error (all data points). Error is the distance from the point to the line of regression.

Consider a dataset containing information on the relationship between' studied number of hours' and' marks received.' Some students were studied and they reported their hours of study and rating. This is going to be our data on preparation. The goal is to develop a model that, given the number of hours studied, can predict marks. A regression line is obtained using the training data, which will result in a minimal error. For any new data, this linear equation is then used. That is, if we give a student as an input number of hours studied, our model will predict their mark with a minimum of error.

$$Y(pred) = b0 + b1 * x \tag{2.2}$$

To minimize the error, the values b0 and b1 must be chosen. If the sum of the squared error to test the model is taken as a metric, then the goal is to obtain a line that best reduces the error [21].

2.8.4 K-nearest neighbor Regression

A simple implementation of K-nearest neighbor (KNN)regression is to calculate the average of the numerical target of the K nearest neighbors. Another approach uses an inverse distance weighted average of the K nearest neighbors. K-nearest neighbor (KNN) regression uses the same distance functions as K-nearest neighbor (KNN) classification.

We have an independent variable (or set of independent variables) and a dependent variable (due to our independent variables we are trying to guess). We might assume, for example, that height is the independent variable and that weight is the dependent variable. Each row is typically referred to as an example, observation, or data point, whereas each column is often referred to as an indicator, element, independent variable, or function (not including the label / dependent variable). Choosing the right K for the data is done by trying several Ks and picking the one that works best [22].

2.8.5 Choice Of Algorithms

Selecting algorithms that are ideal for any research is crucial and not an easy decision to make. But there are algorithms that are suitable for certain topics than others by earlier documented performances. Choosing algorithms that have worked successfully in earlier research's and also considering an improvement in the research, Random Forest Regression, Ridge Regression, Linear Regression and K-nearest neighbor Regression algorithms were selected for this research in order to yield desirable results.

Chapter 3

Related Work

Mascharka and Manley [23] used Random forest, K- Nearest Neighbor (KNN) and Linear Regression among other machine learning algorithms and examined their performances for indoor positioning using a dataset of 3110 datapoints. Fingerprinting approach was used to collect the data which was collected at the Cowles Library at Drake University. Random forest and K- Nearest Neighbor (KNN) were among the most accurately performing algorithms while Linear Regression performed averagely better out of the twenty machine learning algorithms that were measured based on their respective mean errors. The performance of the machine learning models that showed an average error of 0.76 were found to be exceeding the performance of the previously implemented models.

Zan Li, Liang, Braun, Zhao and Xiaohui [24] proposed an indoor positioning system with a fingerprinting technique which combines time information and received signals strengths(RSSI). Random forest regression model is implemented to predict the location of the source and helped for accurate pattern matching. K- Nearest Neighbor (KNN) algorithm is implemented to match the target device position with the RSSI map. A new model KNN-RF is proposed by fusing both the models which shows a significant improvement in positioning accuracy by 36.1 than the previously implemented RSS based fingerprinting. The performance of the proposed models showed a mean position accuracy of 1.61m.

Takayama, Umezawa, Komuro and Osawa [25] evaluated an indoor positioning system using fingerprinting method for collecting radio fingerprints using 16 beacons. A student room was used for collecting the Received Signal Strength Indicator (RSSI) values. Random Forest regression model was used to estimate the position of the location fingerprints. The aim of the research was to compare the proposed method with a baseline method which uses Random Forest regression model to estimate the location. The model was evaluated using 10 fold cross validation and Root Mean Squared Error(RMSE) was used as the performance metric for estimating the distance error and the output was represented by Cumulative Distribution Function(CDF) graphs. The root mean square error of position estimation was 0.87 m.

Bekkelien, Anja, Michel Deriaz, and Stéphane Marchand-Maillet [13] conducted an indoor positioning system based on Bluetooth RSSI. Naïve Bayes' Classier, k-NN and k-NN regressor were implemented to fingerprint method and the positioning algorithms were evaluated based on precision and accuracy. The accuracy was evaluated by the mean distance error between the real position and the estimated one. Precision is checked by discrete values from a cumulative probability functions. The results showed that the k-NN regression algorithm gave the best accuracy and precision of the three.

Khuong and Luo [26] proposed a system which used fingerprinting approach to collect the RSSI signals and used k-NN, Ridge regression models to perform the location prediction. Two datasets were used to predict the location and the error shows 1.5 m in some cases and 1 m in other. CDF plot was used to plot the error of euclidean error. 10 fold cross validation was performed in order to validate the results which show ridge regressor performing slightly better than K-nn regressor.

Alexander and Kusuma [27] proposed a system for indoor positioning using BLE beacons undertaking the fingerprinting approach to collect data and use Random Forest Regression, Linear Regression models for predicting the location. Indoor positioning was done in two phases online and offline. The location prediction is done by predicting the users location in the form of X and Y coordinates. Four beacons were used for data collection and 244 data points were collected and trained with four fold cross validation. Euclidean distance error is used to calculate the error between actual user location (X, Y) and predicted user position (X Y) and RMSE. The results show that Linear Regression model performed better than Random forest model while calculating the mean error results.

Feng, Cao and Yan [28], proposed a system using Ridge regression for effective indoor positioning based on Fingerprinting approach. The experiment is conducted in two phases online and offline, both of which yielded a reduce the deviation of the output weight estimation and the ridge regression model obtains the final position. The off-line training and on-line test data were 1550 and 713. A CDF plot shows the performance of the proposed algorithm which is significantly improves the compared one. The ridge regressor obtained has better generalization ability, because the ridge parameter is obtained with the variance of the training error.

Chapter 4
Method

The methodology used for the research of this thesis are **Literature review** and **Experiment**.

The research questions RQ1 will be addressed by conducting a thorough literature review. An experiment will be performed to address the final research question RQ2.

4.1 Literature review

The guidelines of Wohlin [29] are followed while performing a literature review by using snowball sampling technique. The different machine learning models that were implemented specifically for indoor positioning are distinguished. The suitable models for this research are chosen based on the specific issue of prediction of location in a healthcare environment. Four regression models and Euclidean distance, MAE, RMSE performance metrics are selected based for the literature review. The articles for conducting a literature review were searching the strings,

- “Indoor positioning using machine learning”,
- “Random Forest Regression for indoor positioning”,
- “Ridge Regression for indoor positioning”,
- “Linear Regression for indoor positioning”,
- “K-nearest neighbor Regression for indoor positioning”
- "Indoor positioning using Random Forest Regression, Ridge Regression, Linear Regression, K-nearest neighbor Regression models”

The articles were extracted from Google scholar, IEEE Xplore. The inclusion and exclusion criteria for the article were a follows:

Inclusion criteria

- Articles using English language.
- Articles which are available in full text.
- Conference and journal articles were chosen specifically.
- Articles published between 2000-2019 were chosen.

Exclusion criteria

- Articles which are not in English.
- Articles excluding the 2000-2019 time frame.

4.2 Experiment

An experiment has the advantage over the other research methods such as case study, survey when quantitative data is involved in the research. Hence, an experiment is chosen to address the final research question, RQ2. The independent variables and dependant variables of this experiment are:

Independent variables: Distance, (x ,y) coordinates, RSSI.

Dependent variables: Positioning error.

4.3 Dataset

The dataset used for this thesis was collected at the offices at Lead. A floor plan was generated by measuring the length of the office space used for the experiment. The dataset consists of seven bluetooth beacons signals(Beacon1- Beacon7) and the location co-ordinates of the reciever(x,y). The beacon signals are RSSI and x and y are co-ordinates. The dataset was collected in the year 2019.

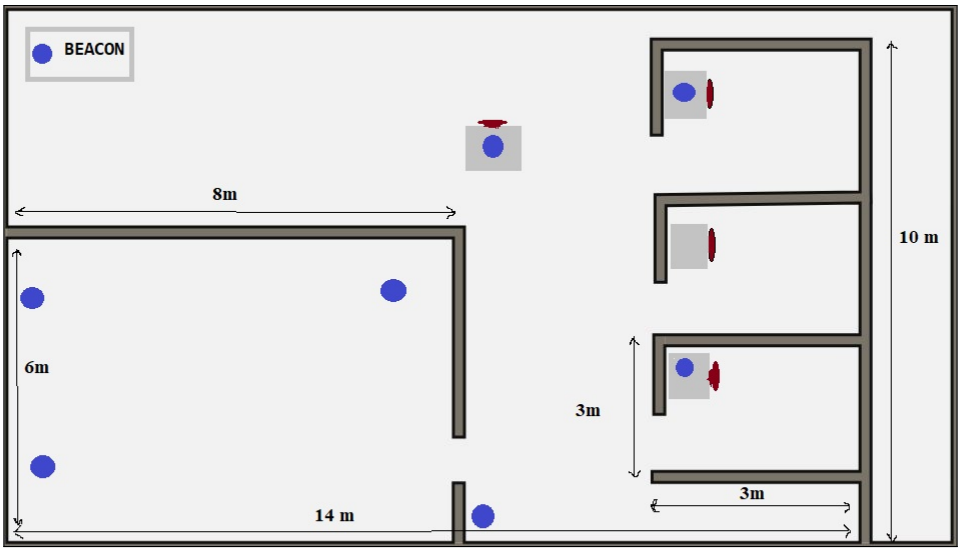


Figure 4.1: Floor Plan showing the Beacons

4.4 Data Preprocessing

The data collected is filtered for removing noise by using running average and if an RSSI is missed from a device, the filter inserts a minimum possible RSSI value in the vacant place. That data is also checked for null values which are also replaced with the minimum possible value of the same.

4.5 Implementation

The experiment was conducted at the first floor of LEAD Incubate at Mjärdevi Science Park. Seven iBeacons and an android smartphone were used to conduct the experiment. The bluetooth beacons were provided by NavAlarm company.

4.5.1 Fingerprinting phase

The Fingerprinting phase is used to collect the data and is carried out by dividing the floor plan into sub areas. Every sub area is called a radio map. By standing on the floor with the smartphone, RSSI values are received from k nearest iBeacons devices are recorded.

A total of n samples are obtained from each iBeacons system at each radio map location. For example, below are the RSSI samples ($R_{map1rssi}$) collected on the radio map (R_{map1}), where ($B1_{rssi1}$) is the first RSSI value from system B1. The data collected is then trained using the selected regression models [14].

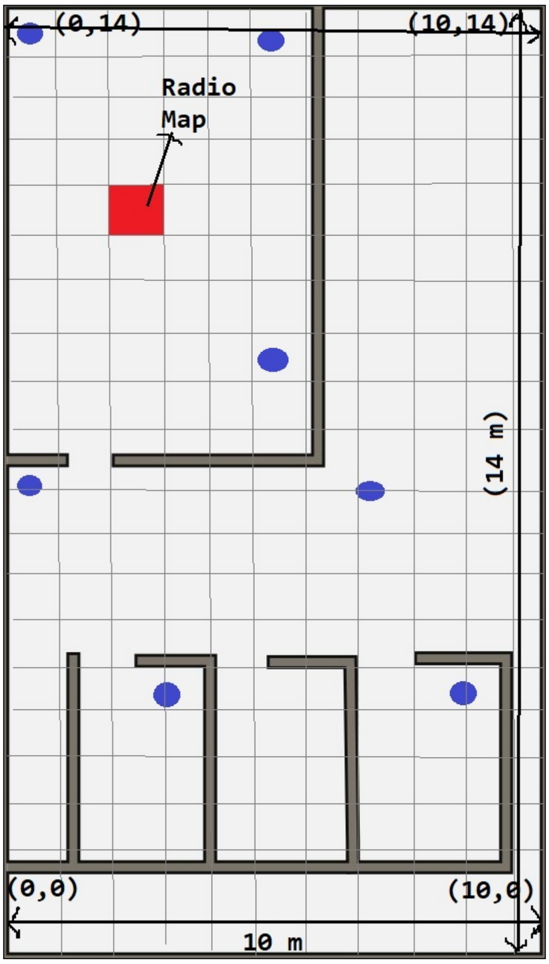


Figure 4.2: Floor Plan showing Fingerprinting phase

$$radiomap_1, radiomap1_{rssi} = \left\{ \begin{matrix} B1_{rssi1} & B2_{rssi1}.... & B7_{rssi1} \\ B1_{rssi2} & B2_{rssi2}.... & B7_{rssi2} \\ & ..... & \\ B1_{rssin} & B2_{rssin}.... & B7_{rssin} \end{matrix} \right\}$$

Every radio map has coordinates (x, y) and is presented as a ($\mathrm{map}_{\mathrm{ix}}$,$\mathrm{map}_{\mathrm{iy}}$). The radio map and RSSI set are used to train the regression models by passing (x,y) coordinates as input to the model. Radio map represents a class and features are represented by the k different RSSI samples from the k iBeacons. The machine learning algorithms return the likelihood of classification of each class given the input features.

Coordinate (x,y)	**class**	**Features**
$(\mathrm{map}_{1x},\mathrm{map}_{1y})$	map_{1}	$\mathrm{map1}_{rssi}$
$(\mathrm{map}_{2x},\mathrm{map}_{2y})$	map_{2}	map_{2rssi}
$(\mathrm{map}_{ix},\mathrm{map}_{iy})$	map_{i}	mapi_{rssi}
$(\mathrm{map}_{mx},\mathrm{map}_{my})$	map_{m}	mapm_{rssi}

Table 4.1: Radio maps with Class, Features and Coordinate

4.5.2 Machine Learning phase

After data collection, we have 1916 data points from seven iBeacons. The machine-learning algorithms, given the 1916 RSSI samples from their respective radio mapping coordinates as input, return the probability of being in each radio map position.

$$RSSI_{samples} + (x, y) => Machine => (p_1, p_2, ...p_m)$$

where, p_i represents the probability of being at the mapith radio map and the the probability of being at each radio map gives the estimated (X,Y).

$$X = \sum_{i=1}^{m} \times map_{\mathrm{ix}} \tag{4.1}$$

$$Y = \sum_{i=1}^{m} \times map_{\mathrm{iy}} \tag{4.2}$$

4.6 Software Environment

4.6.1 Python

Python is a common programming language because of its simplicity, ease of use, open source licensing, and accessibility. Python can be used to create a broad range of application from Web, Desktop GUI-based programs, applications to scientific/math programs, and Machine Learning. In this thesis, different programming features of python were used to conduct an experiment. In this dissertation, the following libraries were used [30].

Pandas: Pandas is an open source library with a Python programming language license for BSD, offering high-performance easy-to-use data structures and data analysis tools. It provides tools for reading and writing data between in-memory data structures and different formats, a quick and effective DataFrame, size mutability etc [31].

Numpy: Numerical Python is an open source Python library, which helps to build multidimensional array objects and perform faster mathematical operations. It includes functions needed for simple and complex mathematics and functions for linear algebra [32].

Matplotlib: Matplotlib is a Python 2D plotting library that generates quality statistics for publications across platforms in a multitude of hardcopy formats and interactive environments [33].

Seaborn: Seaborn is a matplotlib-based Python data visualization library. It offers a high-level interface for informative statistical graphics [34].

Sklearn: Sklearn is an open source library with a Python programming language license for BSD built on NumPy, SciPy, and matplotlib.[35] It offers tools for data mining and data analysis.

The machine learning models using Sklearn in this thesis are:

- Random Forest Regression
- Ridge Regression
- Linear Regression
- K-nearest neighbor Regression

The performance metrics imported by sklearn are,

- MAE, RMSE, Euclidean distance.

4.6.2 Jupyter Notebook

Jupyter Notebook is an open-source web application that enables you to generate and share live code, equations, visualizations, and narrative text records. Its uses include, data cleaning and transformation, numerical simulation, statistical modelling, data visualization, machine learning, and much more. In this thesis, Jupyter Notebook was used for writing the code for the machine learning models [36].

4.7 Experimental setup

- The experiment was performed by conducting 5-fold cross validation with Random Forest Regression model, Ridge Regression model, Linear Regression model and K-nearest neighbor Regression models.
- The performance of the algorithms were experimented and the results are compared for selecting the best algorithm for this dataset.

4.8 Performance metrics:

A regression model's efficiency can be understood by understanding the error rate of the model predictions. By understanding how well the regression lines match the data set, you can measure the overall performance. An excellent regression model is one where the distinction between the actual or observed values and the predicted values to train, validate and test data sets is small and unbiased.

Euclidean Distance Error: Euclidean distance is the distance between two points in any number of dimensions - the square root of the sum of the squares of the differences between the respective coordinates in each of the dimensions [37].

$$EuclideanDistance = \sqrt{(x-X)^2 + (y-Y)^2} \tag{4.3}$$

MAE(Mean Absolute Error): It is the sum of absolute differences between the actual and predicted values it doesn't take care of direction that is positive or negative, all the predicted values are pushed towards positive [38].

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

Where 'n' represents the feature variables, 'yi' represents actual values and $\widehat{yi}$ represents the predicted value.

RMSE (Root Mean Square Error): This measures the deviation from the actual values. It calculates the rooted mean of the square errors, i.e. the differences between individual prediction and actual are squared and summed together, which is then squarely rooted and finally divided. Reduce the RMSE value to have better prediction [39].

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

4.9 Cross Validation

Cross-validation is a re-sampling process that is used on a limited data sample to evaluate machine learning models.The procedure has a single parameter k, which refers to the number of groups to be divided into by a given data sample. The procedure is often referred to as k-fold cross-validation as such. When selecting a specific value for k, it can be used in the model reference instead of k, like k=5 becoming a 5-fold cross-validation [40].
Here the set of data is divided into five folds. The first fold is used to test the model in the first iteration and the rest is used to train the model. Second fold is used as the test set in the second iteration, while the rest serve as the training set. This process is repeated until the test set has been used for each fold of the 5 folds [41].

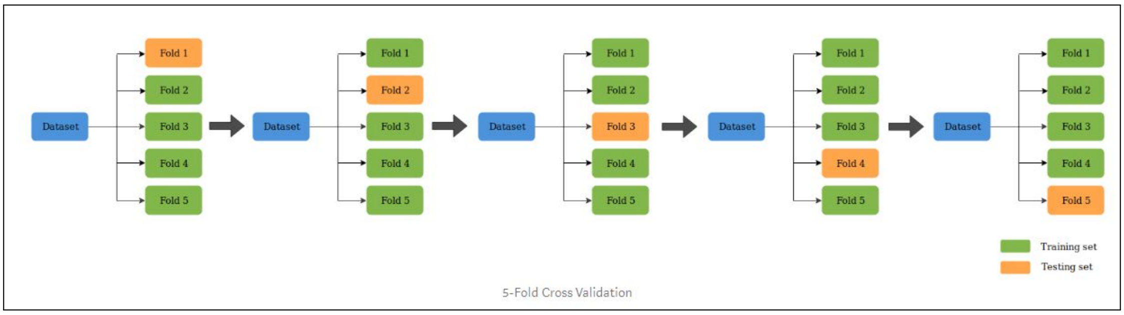
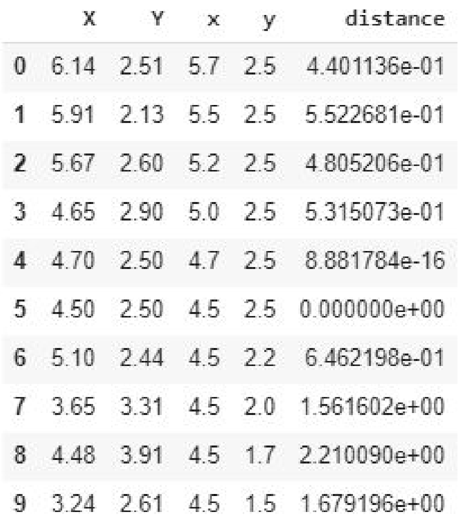


Figure 4.3: K-Fold Cross Validation

Chapter 5

Results

5.1 Random Forest Regression



	X	Y	x	y	distance
0	6.14	2.51	5.7	2.5	4.401136e-01
1	5.91	2.13	5.5	2.5	5.522681e-01
2	5.67	2.60	5.2	2.5	4.805206e-01
3	4.65	2.90	5.0	2.5	5.315073e-01
4	4.70	2.50	4.7	2.5	8.881784e-16
5	4.50	2.50	4.5	2.5	0.000000e+00
6	5.10	2.44	4.5	2.2	6.462198e-01
7	3.65	3.31	4.5	2.0	1.561602e+00
8	4.48	3.91	4.5	1.7	2.210090e+00
9	3.24	2.61	4.5	1.5	1.679196e+00

Figure 5.1: Random Forest Regression Predictions

Random Forest Regressor is trained with the dataset of 1916 data points by using 5-fold cross-validation approach. Where 80% of the data was used for training and 20% of the data, i.e 384 data points was used as the test set. A sample set showing 10 predictions (X,Y) are mentioned in the Figure 5.1. 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 (X , Y) are estimated coordinates of the location. Random Forest Regressor gave a Mean Euclidean Distance error of 3.7.

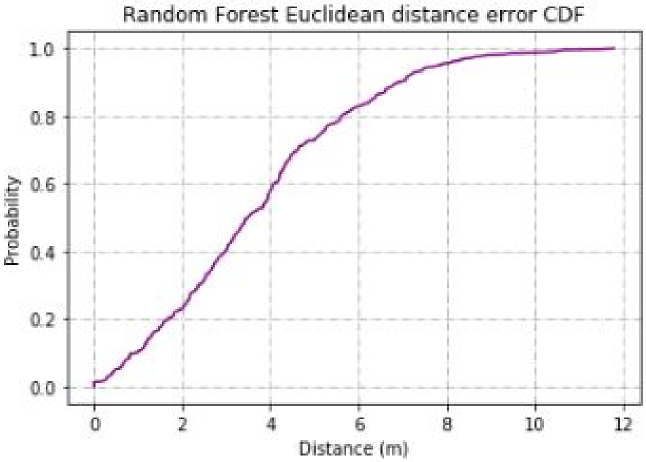
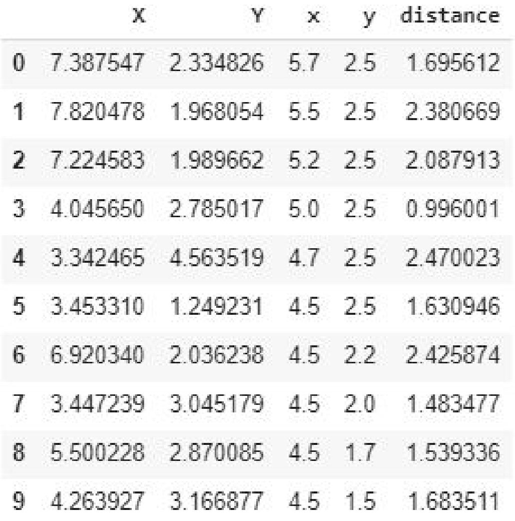


Figure 5.2: Random Forest Regression CDF graph

Figure 5.2 shows the Cumulative Distribution Function (CDF) graph of the Random Forest Regression model. From the CDF graph, Random Forest shows an 80 percentile location error of less than 6 m.

5.2 Ridge Regression



	X	Y	x	y	distance
0	7.387547	2.334826	5.7	2.5	1.695612
1	7.820478	1.968054	5.5	2.5	2.380669
2	7.224583	1.989662	5.2	2.5	2.087913
3	4.045650	2.785017	5.0	2.5	0.996001
4	3.342465	4.563519	4.7	2.5	2.470023
5	3.453310	1.249231	4.5	2.5	1.630946
6	6.920340	2.036238	4.5	2.2	2.425874
7	3.447239	3.045179	4.5	2.0	1.483477
8	5.500228	2.870085	4.5	1.7	1.539336
9	4.263927	3.166877	4.5	1.5	1.683511

Figure 5.3: Ridge Regression Predictions

Ridge Regressor is trained with the dataset by using 5-fold cross-validation approach. Where 80% of the data was used for training and 20% of the data, i.e 384 data points was used as the test set. A sample set showing 10 predictions (X,Y) are mentioned in the Figure 5.3. 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 (X,Y) are estimated coordinates of the location. Ridge Regressor gave a Mean Euclidean Distance error of 4.3.

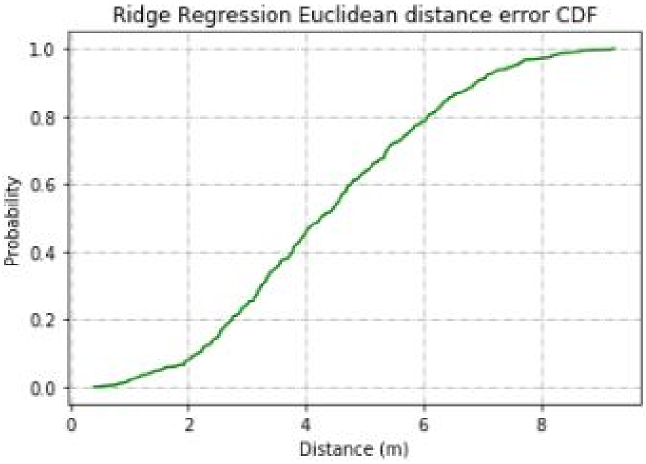
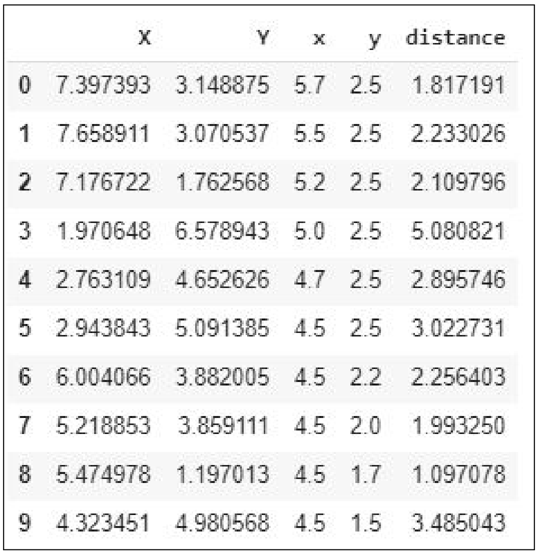


Figure 5.4: Ridge Regression CDF graph

Figure 5.4 shows the Cumulative Distribution Function (CDF) graph of the Ridge Regression model. From the CDF graph, Ridge Regression shows an 80 percentile location error of 6 m.

5.3 Linear Regression



	X	Y	x	y	distance
0	7.397393	3.148875	5.7	2.5	1.817191
1	7.658911	3.070537	5.5	2.5	2.233026
2	7.176722	1.762568	5.2	2.5	2.109796
3	1.970648	6.578943	5.0	2.5	5.080821
4	2.763109	4.652626	4.7	2.5	2.895746
5	2.943843	5.091385	4.5	2.5	3.022731
6	6.004066	3.882005	4.5	2.2	2.256403
7	5.218853	3.859111	4.5	2.0	1.993250
8	5.474978	1.197013	4.5	1.7	1.097078
9	4.323451	4.980568	4.5	1.5	3.485043

Figure 5.5: Linear Regression Predictions

Linear Regression is trained with the dataset by using 5-fold cross-validation approach. Where 80% of the data was used for training and 20% of the data, i.e 384 data points was used as the test set. A sample set showing 10 predictions (X,Y) are mentioned in the Figure 5.5. 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 (X, Y) are estimated coordinates of the location. Linear Regression gave a Mean Euclidean Distance error of 4.4.

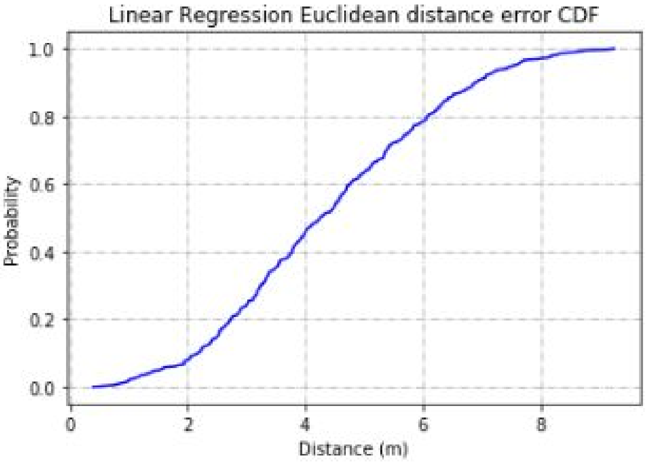
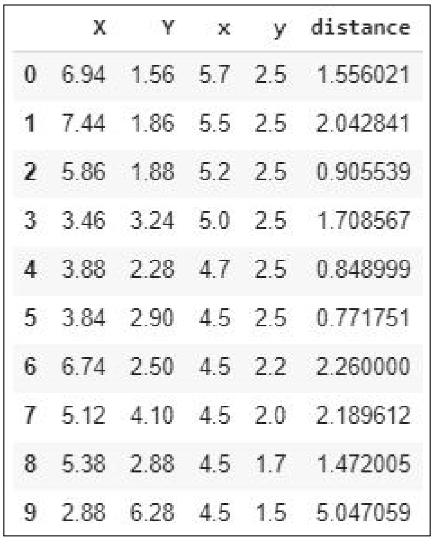


Figure 5.6: Linear Regression CDF graph

Figure 5.6 shows the Cumulative Distribution Function (CDF) graph of the Linear Regression model. From the CDF graph, Linear Regression shows an 80 percentile location error of 6 m.

5.4 K-nearest neighbor Regression



	X	Y	x	y	distance
0	6.94	1.56	5.7	2.5	1.556021
1	7.44	1.86	5.5	2.5	2.042841
2	5.86	1.88	5.2	2.5	0.905539
3	3.46	3.24	5.0	2.5	1.708567
4	3.88	2.28	4.7	2.5	0.848999
5	3.84	2.90	4.5	2.5	0.771751
6	6.74	2.50	4.5	2.2	2.260000
7	5.12	4.10	4.5	2.0	2.189612
8	5.38	2.88	4.5	1.7	1.472005
9	2.88	6.28	4.5	1.5	5.047059

Figure 5.7: K-nearest neighbor Regression Predictions

K-nearest neighbor is trained with the dataset by using 5-fold cross-validation approach. Where 80% of the data was used for training and 20% of the data, i.e 384 data points was used as the test set. A sample set showing 10 predictions (X,Y) are mentioned in the Figure 5.7. 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 (X , Y) are estimated coordinates of the location. K-nearest neighbor gave a Mean Euclidean Distance error of 4.2. Figure 5.8 shows the Cumulative Distribution

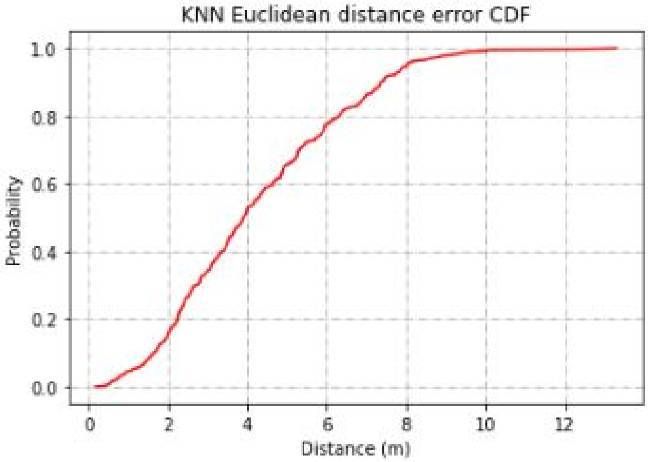


Figure 5.8: K-nearest neighbor Regression CDF graph

Function (CDF) graph of the K-nearest neighbor Regression model. From the CDF graph, K-nearest neighbor shows an 80 percentile location error of more than 6 m.

Chapter 6

Analysis

6.1 Comparative study of Performance Metrics

Algorithms	**Mean Absolute Error**	**Root Mean Square Error**
Random Forest Regression	2.409	3.133
Ridge Regression	2.816	3.370
Linear Regression	2.719	3.352
K-NN Regression	2.696	3.401

Table 6.1: Comparison of performance evaluation results

From the table 6.1, Random Forest Regression performed well with both the metrics MAE and RMSE. Random Forest Regression has shown the minimum error in location estimation when compared to the Ridge Regression, Linear Regression and K-nearest neighbor Regression. Ridge Regressor and K-nearest neighbor Regressor demonstrated the highest error in MAE and RMSE metrics respectively.

6.2 Mean Absolute Error

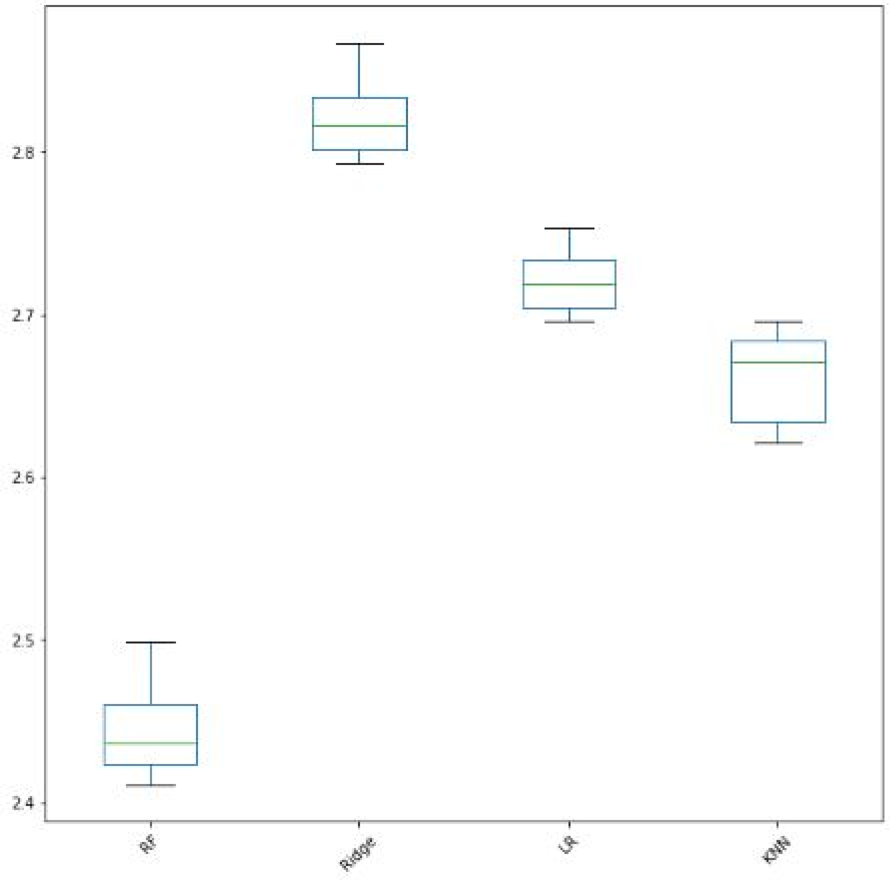


Figure 6.1: Box plot comparison of MAE obtained by regression models

Figure 6.1 represents the Mean Absolute error from the results of the predictions produced by the Random Forest Regression, Ridge Regression, Linear Regression and K-nearest neighbor Regression on 5-fold cross-validation tests. From the figure, it can be seen that Random Forest Regressor has minimum MAE(Mean Absolute Error) when compared to other models. Ridge Regressor has highest MAE and thus it can be said as worst performer.

6.3 Root Mean Squared Error

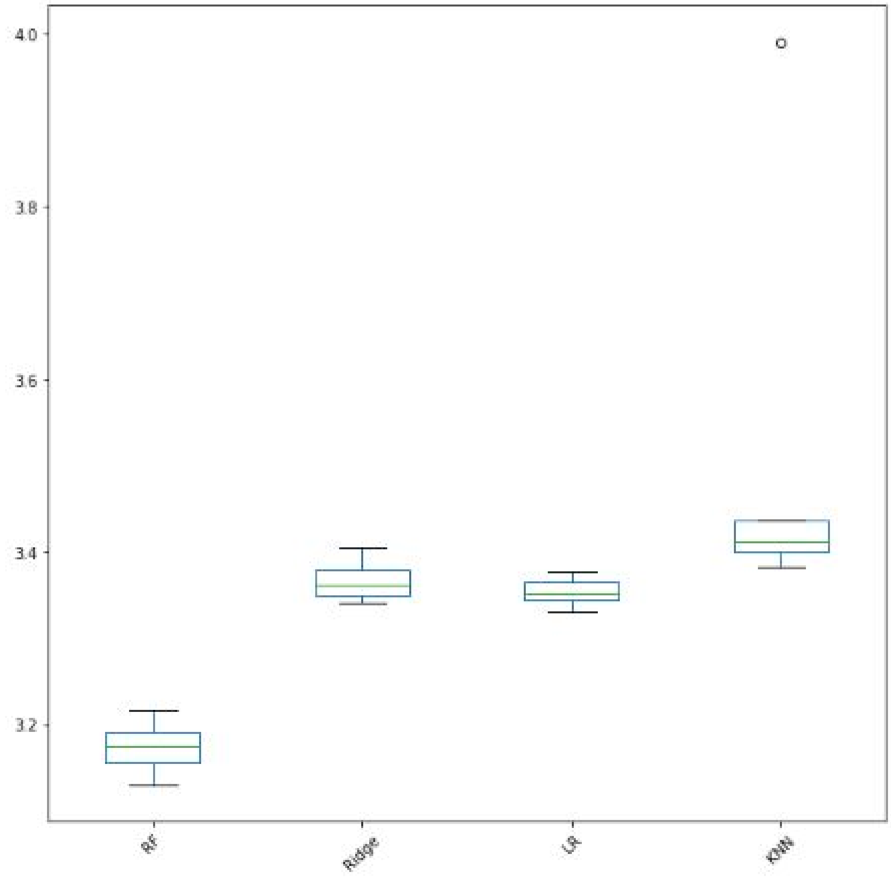


Figure 6.2: Box plot comparison of RMSE obtained by regression models

Figure 6.2 represents the Root Mean Squared Error from the results of the predictions produced by the Random Forest Regression, Ridge Regression, Linear Regression and K-nearest neighbor Regression on 5-fold cross-validation tests. From the figure, it can be seen that Random Forest regressor has minimum RMSE (Root Mean Squared Error) when compared to other models. K-nearest neighbor regressor has highest RMSE and thus it can be said as worst performer.

6.4 Key Analysis

- Random Forest showed better results in RMSE and MAE than the other models. In this research, indoor positioning requires multiple output results in the form of (X,Y) coordinates. Random Forest model achieves this goal better than other models since it is easier hyper parameter tuning which resulted in less error when compared to other models. It also showed the least mean euclidean error.
- Ridge Regression has performed badly in RMSE and MAE when compared to other models because of using a lower dimensional dataset. Ridge model uses regularization which leads to dimensionality reduction which is the possible reason for a high bias error. It showed high mean euclidean error when compared to other models
- Linear Regression being sensitive to outliers and noises in data is prone to overfitting of the data and hence performed badly when compared to Random Forest Regression and K-nearest neighbor. It showed the highest mean euclidean error when compared to other models
- K-nearest neighbor produced better results in MAE when compared to other models but gave bad results in RMSE, because of the amount of samples being small for the model and since it is a slow model to learn. It gave the second least mean euclidean error.

6.5 Discussion

RQ1. What are the suitable machine learning techniques to address indoor positioning using BLE beacons?
Answer: Based on the results obtained from the Literature review, four machine learning models namely Random Forest Regressor, Ridge Regressor, Linear Regressor and K-nearest neighbor Regressor have been chosen for indoor positioning using BLE beacons.

RQ2. What are the performances of machine learning models for location estimation based on the collected data?
Answer: Random Forest Regression showed the least error in RMSE, MAE . The reason for this is because the output of this research for indoor positioning requires multiple coordinates as output which is easier for tuning with the hyper parameters of the random forest model. The Mean euclidean distance error of Random Forest model is the lowest localization error which is 3.7 and an 80 percentile location error in CDF graph of less than 6 m. The highest mean euclidean distance error is shown by Linear Regression, which is 4.4. Both Linear

Regression and Ridge regression showed a an error of 6 m in the 80 percentile location error in CDF graphs. The Average RMSE of Random Forest Regression across the 5-fold cross validation is 3.133 and Average MAE is 2.409. K-nearest neighbor Regressor has shown worst performance of Average RMSE when compared to other models across the 5-fold cross-validation which is 3.401. It also showed an 80 percentile location error in CDF graph of more than 6 m. Ridge Regressor has the worst performance of Average MAE when compared to other models across the 5-fold cross validation which is 2.816. The performances of all the models have been discussed in Section 6.1

6.6 Threats to Validity

Validity is an indicator of how well an assessment really tests what it should be measuring [42].

6.6.1 Internal Validity

The threat of less sample size was rectified by extracting more data.To overcome the threat of missing observations in the experiments, cloud backup is used which consists of all the logs copies of the experiment

6.6.2 External Validity

The threat of less processing power was validated for using external servers with higher computation power. The threat of time taken to run the models was validated by running the models as a batch.

6.7 Limitations

- Having a proper amount of data is essential to properly train and evaluate the machine learning models. The research was conducted with a dataset of with less data points. The results might have been affected by this. To get a valid research result, it is necessary to have a dataset with more data points when conducting a thesis. More data would have enabled further experiments to be conducted, producing results that could increase the study's credibility or highlight the method's problems. The larger the dataset the more reliable will be the results.

- To get maximum prediction accuracy, each particular problem in machine learning uses different hyper parameters, even if the underlying model is the same. Seeing as this is the case, finding the optimal parameters could be difficult. One possibility is to be inspired in similar settings by other models.

Chapter 7

Conclusions and Future Work

In this research, Random Forest Regression, Ridge Regression, Linear Regression and K-nearest neighbor Regression algorithms are identified as suitable machine learning algorithms and they are trained with a dataset containing RSSI values from BLE beacons which were used for conducting the experiment for indoor positioning to track patients and hospital equipment in a healthcare environment. The location prediction is aimed to help reduce the time and effort needed by the staff which can be used to examine other patients. Using performance metrics, Euclidean Distance error, RMSE and MAE for error estimation, the trained algorithms were assessed on 1916 data points. After analysing the results, it is found that Random Forest Regression showed better prediction results in MAE and RMSE when compared to that of Ridge Regression, Linear Regression and K-nearest neighbor Regression algorithms. Ridge regression gave the worst MAE and K-nearest neighbor gave the worst RMSE respectively. The CDF graphs show Random forests 80 percentile location error of less than 6 m and K-nearest neighbors 80 percentile location error of more than 6 m whereas Ridge Regressor and Linear Regressor, showed an 80 percentile location error of 6 m . However, since this thesis is aimed to research a real-world issue, it can be concluded that based on performing better in two out of three metrics, Random Forest is the algorithm of choice for predicting location in this research and in a healthcare environment. The future research can be done by implementing moving beacons and using more advanced machine learning algorithms.

References

[1] Luca Calderoni, Matteo Ferrara, Annalisa Franco, and Dario Maio. Indoor localization in a hospital environment using random forest classifiers. *Expert Systems with Applications*, 42(1):125–134, 2015.

[2] Xin-Yu Lin, Te-Wei Ho, Cheng-Chung Fang, Zui-Shen Yen, Bey-Jing Yang, and Feipei Lai. A mobile indoor positioning system based on ibeacon technology. In *2015 37th Annual International Conference of the IEEE Engineering in Medicine and Biology Society (EMBC)*, pages 4970–4973. IEEE, 2015.

[3] Ramsey Faragher and Robert Harle. Location fingerprinting with bluetooth low energy beacons. *IEEE journal on Selected Areas in Communications*, 33(11):2418–2428, 2015.

[4] Yankai Wang, Qingyu Yang, Guangrui Zhang, and Peng 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)*, pages 243–247. IEEE, 2016.

[5] Pavel Kriz, Filip Maly, and Tomas Kozel. Improving indoor localization using bluetooth low energy beacons. *Mobile Information Systems*, 2016, 2016.

[6] Ramsey Faragher and Robert Harle. An analysis of the accuracy of bluetooth low energy for indoor positioning applications. In *Proceedings of the 27th International Technical Meeting of The Satellite Division of the Institute of Navigation (ION GNSS+ 2014)*, volume 812, pages 201–210, 2014.

[7] Hiroyuki Torii, Shinsuke Ibi, and Seiichi Sampei. Indoor positioning and tracking by multi-point observations of ble beacon signal. In *2018 15th Workshop on Positioning, Navigation and Communications (WPNC)*, pages 1–5. IEEE, 2018.

[8] Faheem Zafari, Athanasios Gkelias, and Kin K Leung. A survey of indoor localization systems and technologies. *IEEE Communications Surveys & Tutorials*, 2019.

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

[10] indoor BLE Admin. Indoor navigation with ble. https://developex.com/blog/indoor-navigation-with-ble/, Nov 2017.

[11] Beacon locator / android. https://appagg.com/android/tools/beacon-locator-18415522.html?hl=en, journal=AppAgg, author=(9), SameBits app, Mar 2019.

[12] Mehdi Mohammadi, Ala Al-Fuqaha, Mohsen Guizani, and Jun-Seok Oh. Semisupervised deep reinforcement learning in support of iot and smart city services. *IEEE Internet of Things Journal*, 5(2):624–635, 2017.

[13] Anja Bekkelien, Michel Deriaz, and Stéphane Marchand-Maillet. Bluetooth indoor positioning. *Master's thesis, University of Geneva*, 2012.

[14] Pranesh Sthapit, Hui-Seon Gang, and Jae-Young Pyun. Bluetooth based indoor positioning using machine learning algorithms. In *2018 IEEE International Conference on Consumer Electronics-Asia (ICCE-Asia)*, pages 206–212. IEEE, 2018.

[15] William L. Hosch. Machine learning. https://www.britannica.com/technology/machine-learning, Oct 2019.

[16] Daphne Koller, Nir Friedman, Sašo Džeroski, Charles Sutton, Andrew McCallum, Avi Pfeffer, Pieter Abbeel, Ming-Fai Wong, David Heckerman, Chris Meek, et al. *Introduction to statistical relational learning*. MIT press, 2007.

[17] What is unsupervised learning? - definition from whatis.com. https://whatis.techtarget.com/definition/unsupervised-learning, journal=WhatIs.com, author=Rouse, Margaret supervised and Haughn, Matthew and Rouse, Margaret and Rouse, Margaret.

[18] Turi machine learning platform user guide. https://turi.com/learn/userguide/supervised-learning/random_forest_regression.html, journal=Random Forest Regression | Turi Machine Learning Platform User Guide, author=for, ran dom.

[19] Turi machine learning platform user guide. https://turi.com/learn/userguide/supervised-learning/regression.html, journal=Regression | Turi Machine Learning Platform User Guide, author=ran, for est.

[20] J Al-Jararha. New approaches for choosing the ridge parameters. *Hacettepe Journal of Mathematics and Statistics*, 47(6):1625–1633, 2016.

[21] Linear regression - detailed view. https://towardsdatascience.com/linear-regression-detailed-view-ea73175f6e86, journal=Medium, publisher=Towards Data Science, author=Swaminathan, Saishruthi lr, year=2019, month=Jan.

[22] Onel KNN Harrison. Machine learning basics with the k-nearest neighbors algorithm. `https://towardsdatascience.com/machine-learning-basics-with-the-k-nearest-neighbors-algorithm`, Jul 2019.

[23] David Mascharka and Eric Manley. Lips: Learning based indoor positioning system using mobile phone-based sensors. In *2016 13th IEEE Annual Consumer Communications & Networking Conference (CCNC)*, pages 968–971. IEEE, 2016.

[24] Zan Li, Torsten Braun, Xiaohui Zhao, Zhongliang Zhao, Fengye Hu, and Hui Liang. A narrow-band indoor positioning system by fusing time and received signal strength via ensemble learning. *IEEE access*, 6:9936–9950, 2018.

[25] Tomofumi Takayama, Takeshi Umezawa, Nobuyoshi Komuro, and Noritaka Osawa. A regression model-based method for indoor positioning with compound location fingerprints. *Geo-spatial Information Science*, pages 1–7, 2019.

[26] Khuong An Nguyen and Zhiyuan Luo. Reliable indoor location prediction using conformal prediction. *Annals of Mathematics and Artificial Intelligence*, 74(1-2):133–153, 2015.

[27] Ivan Alexander and Gede Putra Kusuma. Predicting indoor position using bluetooth low energy and machine learning.

[28] Zhiyue Feng, Yanhua Cao, and Jun Yan. A received signal strength based indoor localization algorithm using elm technique and ridge regression. In *2019 IEEE 2nd International Conference on Electronic Information and Communication Technology (ICEICT)*, pages 599–603. IEEE, 2019.

[29] Claes Wohlin. Guidelines for snowballing in systematic literature studies and a replication in software engineering. In *Proceedings of the 18th international conference on evaluation and assessment in software engineering*, page 38. Citeseer, 2014.

[30] Chris python Kambala. Python programming for data science and machine learning - dzone ai. `https://dzone.com/articles/python-programming-for-data-science-and-machine-le`, Jun 2019.

[31] python pandas pydata. Python data analysis library. `https://pandas.pydata.org/`.

[32] Numpy. `https://numpy.org/`.

[33] Documentation. `https://matplotlib.org/`.

[34] seaborn. statistical data visualization. https://seaborn.pydata.org/.

[35] Scikitlearn. https://scikit-learn.org/stable/.

[36] Jupyternotebook. https://jupyter.org/.

[37] Christoph Euclidean distance Ruegg, Marcus Cuda, and Jurgen Van Gael. https://numerics.mathdotnet.com/Distance.html.

[38] E M. Mean absolute error mae machine learning(ml). https://medium.com/@ewuramaminka/mean-absolute-error-mae-machine-learning-ml-b9b4afc63077, Feb 2018.

[39] RMSE Stephanie. Rmse: Root mean square error. https://www.statisticshowto.datasciencecentral.com/rmse/, Oct 2019.

[40] Jason Brownlee. A gentle introduction to k-fold cross-validation. https://machinelearningmastery.com/k-fold-cross-validation/, Aug 2019.

[41] Payam Refaeilzadeh, Lei Tang, and Huan Liu. Cross-validation. *Encyclopedia of database systems*, pages 532–538, 2009.

[42] Irena Nančovska Šerbec, Mateja Strnad, and Jože Rugelj. Assessment of wiki-supported collaborative learning in higher education. In *2010 9th International Conference on Information Technology Based Higher Education and Training (ITHET)*, pages 79–85. IEEE, 2010.