

Development of model based on nearest neighbours with previous knowledge of signal propagation

**Vicenç Pio Badia**

Master ciència de dades

Area 4??

**Nom Consultor/a**

**Joaquín Torres-Sospedra**

Data Lliurament

Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](http://creativecommons.org/licenses/by-nc-nd/3.0/es/)



**Llicències alternatives (triar alguna de les següents i substituir la de la pàgina anterior)**

**A) Creative Commons:**

Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-SenseObraDerivada 3.0 Espanya de Creative Commons](http://creativecommons.org/licenses/by-nc-nd/3.0/es/)



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial-CompartirIgual 3.0 Espanya de Creative Commons](http://creativecommons.org/licenses/by-nc-sa/3.0/es/)



Aquesta obra està subjecta a una llicència de [Reconeixement-NoComercial 3.0 Espanya de Creative Commons](http://creativecommons.org/licenses/by-nc/3.0/es/)



Aquesta obra està subjecta a una llicència de [Reconeixement-SenseObraDerivada 3.0 Espanya de Creative Commons](http://creativecommons.org/licenses/by-nd/3.0/es/)



Aquesta obra està subjecta a una llicència de [Reconeixement-CompartirIgual 3.0 Espanya de Creative Commons](http://creativecommons.org/licenses/by-sa/3.0/es/)



Aquesta obra està subjecta a una llicència de [Reconeixement 3.0 Espanya de Creative Commons](http://creativecommons.org/licenses/by/3.0/es/)



**B) GNU Free Documentation License (GNU FDL)**

Copyright © ANY EL-TEU-NOM.

Permission is granted to copy, distribute and/or modify this document under the terms of the GNU Free Documentation License, Version 1.3 or any later version published by the Free Software Foundation; with no Invariant Sections, no Front-Cover Texts, and no Back-Cover Texts.

A copy of the license is included in the section entitled "GNU Free Documentation License".

**C) Copyright**

© (l'autor/a)

Reservats tots els drets. Està prohibit la reproducció total o parcial d'aquesta obra per qualsevol mitjà o procediment, compresos la impressió, la reprografia, el microfilm, el tractament informàtic o qualsevol altre sistema, així com la distribució d'exemplars mitjançant lloguer i préstec, sense l'autorització escrita de l'autor o dels límits que autoritzi la Llei de Propietat Intel•lectual.

**FITXA DEL TREBALL FINAL**

|  |  |
| --- | --- |
| **Títol del treball:** | *Development of model based on nearest neighbours with previous knowledge of signal propagation* |
| **Nom de l’autor:** | *Vicenç Pio Badia* |
| **Nom del consultor/a:** | *Joaquín Torres-Sospedra* |
| **Nom del PRA:** |  |
| **Data de lliurament (mm/aaaa):** | *06/2022* |
| **Titulació o programa:** | *Màster ciència de dades* |
| **Àrea del Treball Final:** | *Àrea 4* |
| **Idioma del treball:** | *Anglès* |
| **Paraules clau** | *Wi-Fi Fingerprinting, indoor geolocation, k-NN* |
| **Resum del Treball (màxim 250 paraules):** *Amb la finalitat, context d’aplicació, metodologia, resultats i conclusions del treball* | |
| En aquest treball s’estudia el Wi-Fi Fingerprinting, que és una tècnica molt popular utilitzada en el geoposicionament en interiors. Està basada en la mesura de la senyal emesa pels punts d’accés Wi-Fi més propers i disponibles. El conjunt d’intensitats detectades pel dispositiu receptor es coneix com a empremta Wi-Fi (Wi-Fi fingerprint) i és molt útil a l’hora de calcular la posició d’aquest objecte. L’algorisme utilitzat per fer aquest càlcul és una adaptació del k-NN, el qual s’han fet diverses propostes de millora en els últims anys per augmentar-ne el rendiment. Per culpa de la pròpia naturalesa de la propagació de les senyals, apareixen problemes d’interferències provocats per obstacles de l’entorn, tant estàtics com mòbils. En el present treball es busca una implementació de l’algorisme que millori al màxim la precisió del posicionament amb el mínim cost computacional d’entrenament del model de predicció. També es busca el millor ajustament de dos paràmetres de l’algorisme k-NN: el valor de la k i la funció de distància. L’avaluació de l’algorisme es duu a terme utilitzant diverses bases de dades heterogènies amb dades d’entrenament i de test per determinar les millors mètriques de distància per a cada una d’elles. Per últim es proposa una solució que consisteix en fer un *ensemble* de les millors mètriques per obtenir la màxima precisió en els resultats. | |
| **Abstract (in English, 250 words or less):** | |
| Wi-Fi Fingerprinting is a widely used technique for indoor geolocation. It is based on the measurement of the signals emitted by the near and available Wi-Fi access points. The set of signals received by the device is known as Wi-Fi Fingerprint and is very useful in terms of calculating the position of this object. The algorithm behind this operation is an adaptation of the k-NN, which has been the focus of several studies in the recent years to improve its performance. Because of the own nature of the signal propagation, some interference problems appear due to the obstacles in the environment, both static and dynamic. The present work seeks an implementation of the algorithm with the aim to maximize the accuracy and minimize the computational cost of training the prediction model. To achieve that goal, two main parameters are adjusted in the k-NN algorithm: the k value and the distance metric. The model is trained and evaluated using several heterogenous databases. Finally, a solution called *ensemble* is proposed to improve the results, it consists of combining the outputs of the best metrics to reduce the positioning error. | |

**Keywords**

Fingerprint

RSS

Indoor geolocation

Coordinates

**List of abbreviations**

**RSS** Received Strength Signal

**Wi-Fi** Wireless Fidelity

**GPS** Global Positioning System

**Contents**

[1. Introduction 1](#_Toc104023715)

[1.1 Context and work justification 1](#_Toc104023716)

[1.2 Main objectives 1](#_Toc104023717)

[- Data cleaning, data filtering and identification of outliers. 1](#_Toc104023718)

[- Select the signals from the most relevant access points. 1](#_Toc104023719)

[- Select the best data representation to explain the conclusions. 1](#_Toc104023720)

[- Select the best distance metric to optimize the results in this context. 1](#_Toc104023721)

[- Implement and train the k-NN model with best accuracy and lowest computational cost. 1](#_Toc104023722)

[1.4 Approach and metodology 2](#_Toc104023723)

[1.5 Calendar planification 2](#_Toc104023724)

[1.6 Summary of obtained products 2](#_Toc104023725)

[1.7 Summary of the chapters 2](#_Toc104023726)

[2. Related work 3](#_Toc104023727)

[2.1 Subareas within the scope of my work 3](#_Toc104023728)

[2.2 Possible applications in one area 3](#_Toc104023729)

[2.3 Similar investigations 3](#_Toc104023730)

[3. Materials and methods 5](#_Toc104023731)

[4. Description of the proposed approach 7](#_Toc104023732)

[5. Experimental results and discussion 10](#_Toc104023733)

[6. Conclusions and future works 16](#_Toc104023734)

[8. Bibliography 17](#_Toc104023735)

[9. Annex 18](#_Toc104023736)

**List of figures**

**¡Error! No se encuentran elementos de tabla de ilustraciones.**

# 1. Introduction

## 1.1 Context and work justification

Many applications need the exact position of users for proper working and companies require this technology more and more on its new products and services. The growing world of the Internet of Things is one of the sectors with the most demand. Geolocation in outdoor spaces is usually easy to resolve by using GPS systems embedded into all smartphones. However, geolocation in indoor environments comes with more limitations because GPS signals are not strong enough inside a building or in narrow spaces such a street with tall buildings surrounding. Another inconvenience is that GPS cannot distinguish at which floor is placed the device.

There are currently several indoor positioning techniques based on different technologies: Wi-Fi, BLE, RFID, etc. In our case of study, we focus on Wi-Fi fingerprinting with k-NN algorithm to create a prediction model using the measurements of the signals emitted by the near and available Wi-Fi access points. Related studies approaching the problem show results with some accuracy errors between the predicted location and the actual location of the device measured. The aim of this work is to implement a k-NN to improve the accuracy and reduce the computational cost when training the model. The focus is on the treatment of the signal propagation conditions and the handling of the errors caused by the present obstacles in the environment.

## 1.2 Main objectives

The main objective of the project is to design and implement a predictive model based on the k-NN algorithm with previous knowledge of the signal propagation. The partial objectives are the following:

* Data cleaning, data filtering and identification of outliers.
* Implement and train the k-NN model
* Compare results with different k values and distance metricd to optimize the results in this context.
* Find the configuration with best accuracy and lowest computational cost.

1.3 Personal motivation

My interest in engineering and telecommunications comes from many years ago when studying at high school. The field of information technologies was my choice when starting the bachelor at university and for the jobs I’ve worked during the last years. The Master of Data Science opened a new window of knowledge for me, and I believe this topic is a good field of study to do a deeper insight. I have already worked in projects involving indoor positioning but using other technologies such as UWB, RFID and RF but never Wi-Fi. Because all this reasons I think this project is a good opportunity for me to finalize this master’s degree.

## 1.4 Approach and methodology

The source of the data that will be used comes from a database filled with Wi-Fi signals collected from indoor spaces. That data includes the RSSI of each of the signals as well as other information. The first step will be to review the state of the art about Wi-Fi Fingerprinting and related recent studies. The next step is the design and implementation of the algorithm: the first stage will be the data preparation including data cleaning and selection of the interesting data from the training data base. Then the algorithm will be tested using different distance metrics to find the solution that fits best. The same process will be done with the data representation. The test data comes from a different source of the training data to simulate the real conditions and evaluate the algorithm. The development of the project will be coded in Python.

## 1.5 Calendar planification

The project is divided in seven stages to separate the main tasks:

* Definition and planification. From 16/02 to 27/02.
* State of the art revision. From 28/02 to 13/03.
* Design and implementation. From 14/03 to 15/05.
* Writing of the thesis (first delivery). From 16/05 to 29/05.
* Writing of the thesis (final delivery). From 30/05 to 05/06.
* Preparation of defense. From 06/06 to 12/06.
* Public defense. From 13/06 to 24/06.

## 

## 1.6 Summary of obtained products

The obtained product is basically a library of functions with the following implementations:

* Calculation of k-NN and position estimation.
* Positioning error of the prediction.
* Average error of positioning errors.
* Distance metrics comparison.
* Ensemble of selected distance metrics.
* ECDF and scatter plots of the positioning errors.

1.7 Summary of the chapters

This project includes five main chapters that are described as follows:

* Related work: first explains the subareas of the project development, then the possible applications and finally presents a list of similar investigations related to the topic of this project.
* Materials and methods: this chapter introduces the data structure used along the project with examples. It also describes the main functions that compose the k-NN algorithm.
* Description of the proposed approach: it is explained the process and methodology carried out during the project. It details the incremental implementation of the algorithm through different stages and the difficulties encountered.
* Experimental results and discussion: this chapter presents the outcomes obtained from the model and does some comparison and interpretation using plots and tables.
* Conclusions and future works: it explains what initial objectives are achieved or not and why. It also revises the methodology and planification followed and finally states some future lines of investigation related to this topic.

# 2. Related work

In this section, we first detail the different subareas contained within the scope of this work and the summary of the main tasks for each of them. Then we review several areas where this project could be applied and give value to possible applications. Finally, we present the state of the art of similar investigations.

## 2.1 Subareas within the scope of my work

The scope of the project includes several subareas that will be approached during the implementation of the algorithm. They can be divided as follows:

* Data cleaning: is the first step when working with large datasets. Includes the selection of the observations of interest and the possible transformations of some data if needed. A good process of data cleaning is fundamental for the performance of the algorithm because it helps to reduce the computational cost of the training.
* Algorithm implementation: consists of selecting the most relevant fingerprints focusing on the signal propagation. It is the main work of the project, and the goal is to implement an “improved” k-NN that predicts the position of the devices with the best accuracy.
* Train and test: a training database will be used to train the model with the best parameters to obtain the best results. The testing of the algorithm will be done with a separate database filled with testing data.

## 2.2 Possible applications in one area

In the recent years, indoor positioning has grown as one of the most demanded technologies by companies working in sectors such as IoT and Industry 4.0. We can find multiple examples of applications for indoor location, from the navigation of hospitals, airports, parking garages and shopping malls, for example, to navigational aids for the blind and visually impaired, targeted advertising, mining, and disaster response.

The use of Wi-Fi for the positioning comes with two main advantages: all these applications can be embedded in an average smartphone since all of them include Wi-Fi as standard. The other advantage is that most of the buildings have their own Wi-Fi access points so there is no need to install extra hardware to use these applications.

## 2.3 Similar investigations

The last part of this section is to present some of the most relevant works of the state of the art in relation with our field of study. All these articles will be useful to compare results with our own work at the end of the project.

Zhenghua Chen et al., (2019) present a local feature-based deep long short-term memory (LF-DLSTM) approach. The local feature extractor attempts to reduce the noise effect and extract robust local features. The DLSTM network is able to encode temporal dependencies and learn high-level representations for the extracted sequential local features.

Xudong Song et al., (2019) implement a novel classification model and a novel positioning model by combining a Stacked Auto-Encoder (SAE) with a one-dimensional Convolutional Neural Network (CNN). The SAE is utilized to precisely extract key features from sparse Received Signal Strength (RSS) data while the CNN is trained to effectively achieve high accuracy in the positioning phase.

Jianwei Niu et al., (2015) implement weighted KNN algorithm to assign different weights to APs and achieve room-level localization. To obtain the absolute coordinate of users, they design a novel MDS algorithm called MDS-C (Multi-Dimensional Scaling with Calibrations) to calculate coordinates of interested locations in the corridor and rooms, where anchor points are used to calibrate absolute coordinates of users.

Hurkan M. Aydin et al., (2021) propose to use feature selection methods along with the K-nearest neighbours (KNN) classification and regression algorithms in order to create a simple and swift location positioning system. The evaluation of various feature selection methods shows that computation times for positioning can be reduced by 75% using feature selection.

Jing, Hao et al., (2014) design a collaborative Wi-Fi fingerprint training (cWiDB) method that enables the system to perform inertial measurement based collaborative positioning or Wi-Fi fingerprinting alternatively according to the current situation. It also reduces the time required for training the fingerprint database. Different database training methods and different training data size are compared to demonstrate the time and data required for generating a reasonable database. Finally, the fingerprint positioning result is compared which indicates that the cWiDB is able to achieve the same positioning accuracy as conventional training methods but with less training time and a data adjustment option enabled.

Mok, Esmond and Cheung, Bernard K. S. (2013) propose a Wi-Fi positioning algorithm based on neural network modelling of Wi-Fi signal patterns. This algorithm is based on the correlation between the initial parameter setting for neural network training and output of the mean square error to obtain better modelling of the nonlinear highly complex Wi-Fi signal power propagation surface. The test results show that this neural network-based data processing algorithm can significantly improve the neural network training surface to achieve the highest possible accuracy of the Wi-Fi fingerprinting positioning method.

Finally, Rui Zhou et al., (2016) present a Wi-Fi fingerprinting algorithm based on Support Vector Machines (SVM), which combines SVM classification and regression to model the unknown relationship. During sampling and training, the indoor area is partitioned to subregions and the nonlinear relationship between signal fingerprints and locations as well as subregions are established. For positioning, SVM classifiers first determine the subregion that the mobile device is in, then SVM regression estimates the exact coordinate on the basis of classification result.

# 3. Materials and methods

This chapter describes the structure of the data and the environment used along the project with details and examples. It also gives introduction to the main methods implemented to compose the k-NN algorithm.

3.1 Test environment

The data used in the experimentation comes from different areas of University Jaume I (València). It corresponds to several rooms of several floors and buildings and the area includes obstacles that can block or disturb the RSS signals. The infrastructure consists of X Wi-Fi Aps installed… +info

3.2 Datasets structure

The data collected is stored in files with CSV format. These are the datasets used in this project:

|  |  |  |  |
| --- | --- | --- | --- |
| **Filename** | **Train** | **Test** | **Total** |
| DSI1 | 1369 | 348 | 1717 |
| DSI2 | 576 | 348 | 924 |
| LIB1 | 576 | 3120 | 3696 |
| LIB2 | 576 | 3120 | 3696 |
| MAN1 | 14300 | 460 | 14760 |
| MAN2 | 1300 | 460 | 1760 |
| SIM0001 | 10710 | 1000 | 11710 |

Each dataset is divided in four separate CSV files following this structure:

* Training coordinates
* Training RSS
* Test coordinates
* Test RSS

On one hand, the coordinates files include: x, y, z, floor, building. For this work, and with the purpose to simplify the computation, the variables floor and building are avoided.

On the other hand, the RSS files include the actual fingerprint, which is a vector of RSS. The vectors of RSS are represented in dB and the value 100 indicates signal not received. The length of the fingerprints varies from one dataset to another. An example of an RSS row could be:

100,-79,100,100,100,100,100,100,100,100,100,100,100,100,-84,100,100,100,100,100,100,100,

100,100,100,100,100,100,100,100,-72,-72,100,-84,100,100,100,100,100,100,100,100,100,100,

100,100,100,100,100,100,100,100,100,100,100,100,100,100,-85,100,100,100,100,100,100,

100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100,100

Notice that most of the values are 100, it indicates that the signal of this particular Wi-Fi AP is not received by the device measuring.

An example of a coordinates row could be:

73.75653190807489,54.63366094694744,3.7,1,1

That can be translated as:

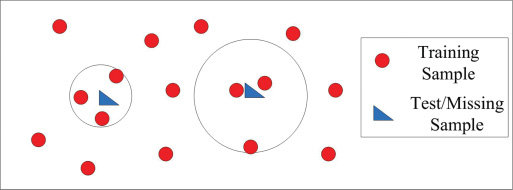
|  |  |
| --- | --- |
| x | 73.75653190807489 |
| y | 54.63366094694744 |
| z | 3.7 |
| floor | 1 |
| building | 1 |

3.3 Methods

In this section are explained the methods implemented during the project:

K-NN

It is a supervised algorithm used for both classification and regression. In our case is used for regression and it consists basically of finding the average value of the k nearest neighbours for each fingerprint to obtain the estimated position coordinates.



Scenario with 2 test samples and k=3

The procedure to find the nearest neighbours follows these steps:

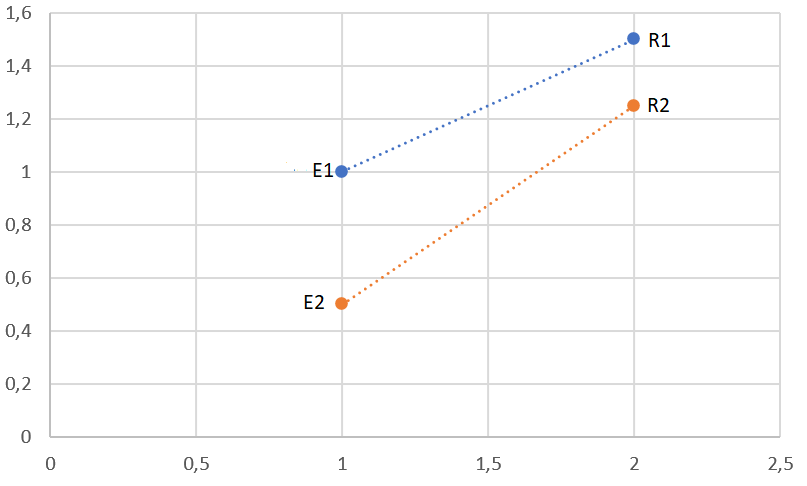
* Calculate the distance between one test sample and all the train samples of the dataset.
* Once obtained the list of distances, the list is sorted and only the first k samples are selected. Those k samples are the k nearest neighbours.
* From these k fingerprints, the next step is to calculate the mean value of their corresponding coordinates.
* The result from the previous calculation is the predicted position of the initial test sample.
* This procedure is repeated for each test fingerprint.

Positioning error

The positioning error is the Euclidean distance between the estimated position and the real position of a sample. The formula of the Euclidean distance in two dimensions is:

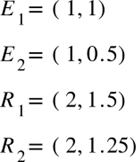


During the development of the project, the evaluation of the algorithm has been based on the average error of the predicted positions. The average error is calculated using the mean value of the positioning errors of all the positions estimated.



Minumum distance between coordinates

In the previous figure is shown an example of two estimated coordinates by the algorithm (E1 and E2) joined with a line to the real position of these coordinates (R1 and R2). The line indicates the Euclidean distance between them. In this case, the coordinates have the value:



The first step is to calculate the distance error of the two pairs of coordinates:



Then the average error is:



Ensemble

It is a technique used to improve the accuracy of the prediction once the k-NN has been executed. It consists of selecting the distance metrics with best accuracy results for one dataset and finding the centroid of the estimated position of each one of them. The output is the final predicted position.

# 4. Description of the proposed approach

The objective of the project is to find the best configuration of the k-NN algorithm to obtain a good accuracy and average error. To achieve that goal, the followed methodology consists of an incremental implementation of the algorithm. That means starting with a basic k-NN on the first tests and then adding several features to end up with the best results. In this chapter is explained this process and the detail of every step.

* Basic k-NN with Euclidean distance: this part is the core of the algorithm. The main functionality is to iterate the fingerprints of the test RSS data to find the k nearest neighbours within the train RSS data for each one of them. This implementation only served to obtain a list of fingerprints indexes, the next step is to link them to a coordinate.
* Predict positions: using the fingerprint indexes, the algorithm finds the corresponding coordinates in the train coordinate dataset and calculates the mean value to end up with the estimated position for each fingerprint.
* Calculate error: the list of predicted positions needs to be contrasted with the actual position stored in the test coordinates dataset. The two lists of positions are compared, and the distance error is calculated using the Euclidean distance. That list of distance errors can be used to evaluate the model using ECDF and scatter plots.
* Average error: another useful metric for evaluation is the average error. This value is obtained calculating the mean of all the distance positioning errors.

At this point, the basic implementation of the algorithm is done. The next steps describe the process followed during the project to optimize the accuracy and reduce the positioning error.

* Distance/similarity metrics comparison: the basic implementation uses the Euclidean distance to find the nearest neighbours. However, it is a good practice to compare the average errors obtained by the execution of the algorithm using different metrics. The list of the metrics studied in this project grouped by families:

|  |  |
| --- | --- |
| **Lp Minkowski family** | |
| Euclidean L2 |  |
| Minkowski Lp |  |
| Chebyshev |  |
| **L1 family** | |
| Sorensen |  |
| Gower |  |
| Soergel |  |
| Kulczynski d |  |
| Lorentzian |  |
| Canberra |  |
| **Intersection family** | |
| Intersection |  |
| Wave Hedges |  |
| Czekanowski s |  |
| Czekanowski d |  |
| Motyka s |  |
| Motyka d |  |
| Kulczynski s |  |
| Ruzicka |  |
| Tanimoto |  |
| **Inner product family** | |
| Inner Product |  |
| Harmonic |  |
| Cosine |  |
| Kumar-Hassebrook |  |
| Jaccard s |  |
| Jaccard d |  |
| Dice s |  |
| Dice d |  |
| **Fidelity family or Squared-chord family** | |
| Fidelity |  |
| Bhattacharrya |  |
| Hellinger |  |
| Matusita |  |
| Squared-chord |  |
| **Squared L2 family** | |
| Squared Euclidean |  |
| Pearson |  |
| Neyman |  |
| Squared |  |
| Probabilistic Symmmetric |  |
| Divergence |  |
| Clark |  |
| Additive symmetric |  |
| **Shannon's entropy family** | |
| Kullbac Leiber |  |
| Jeffreys |  |
| K Divergence |  |
| Topsoe |  |
| Jensen-Shannon |  |
| Jensen difference |  |
| **Combinations** | |
| Taneja |  |
| Kumar Johnson |  |
| AvgL |  |
| **Vicissitude** | |
| Vicis-Wave Hedges |  |
| Vicis Symmetric 1 |  |
| Vicis Symmetric 2 |  |
| Vicis Symmetric 3 |  |
| Min-Symmetric |  |
| Max-Symmetric |  |

As a result of the execution of the different metrics, some errors appeared because of the format of the input data. These errors were always due to divisions by 0 and square root and logarithm of negative numbers. The specific implementation of some of the metrics was incompatible with the format of the fingerprints. To fix this issue, three alternative solutions are proposed: min-max normalization, absolute value and increment:

* Min-max normalization: the concept of min-max normalization stands for finding the lowest and greatest value of the entire dataset and set them to 0 and 1 respectively. From that point, the rest of the values are scaled so that the whole dataset ranges from 0 to 1.

It is important to mention that the scaler must use the RSS values of train and test altogether to find the general minimum and maximum. Otherwise, the scale would be wrong, and that is something that happened to the present implementation and produced some incorrect outputs before it was correctly solved.

* Absolute error: this one is the simplest alternative and allows to convert all the values to positive numbers.
* Increment: this solution involved making the decision of what value choose for the increment. The datasets used has slightly different range of values but, in general, the approximate minimum value is about -95. Based on that, the increment value was set to 100 to ensure all data to be above 0.

The last part of the………

* Ensemble: the ensemble is the final improve made to achieve the objective of optimize the k-NN results. It consists of selecting the best performing metrics and calculate the centroids of the predicted positions. The strength of this procedure is that it combines several results and the possible high errors of one metric prediction are compensated by the others. It could happen that some particular predictions of the ensemble were worse than the original prediction from the metric, but the average error should be better.

# 5. Experimental results and discussion

This chapter exposes the results obtained and gives interpretation and discussion on these.

As previously said, the k-NN algorithm has been tested on seven distinct datasets. The hyperparameters used to test the accuracy are the k value, the distance metric, and the transformation of the fingerprint (normalization, absolute value and increment).

When running the basic k-NN, errors appeared because of divisions by 0, sqrt and log of negative numbers. To avoid these errors, we tried different solutions:

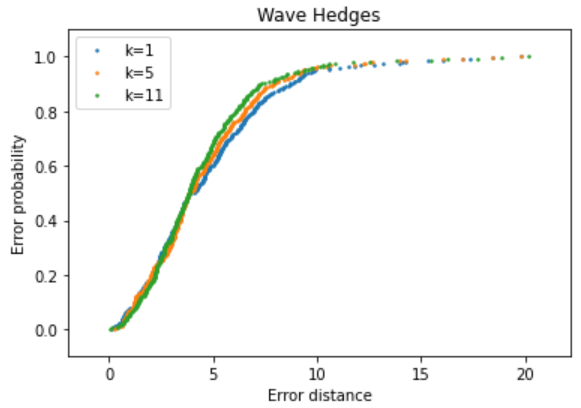
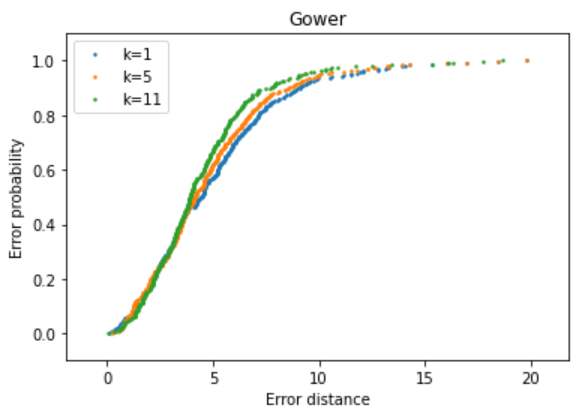
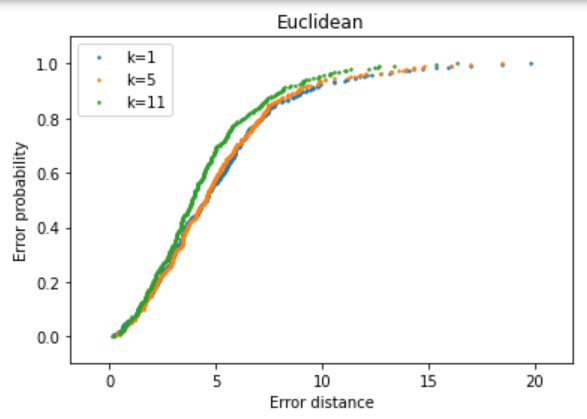
* Min-max normalization:
* Absolute value:
* Increment by 100:

I ran several tests using the default configuration and the three solutions explained above, the best results are obtained with the following configuration:

* Dataset DSI1 average error with **absolute value**:

|  |  |  |  |
| --- | --- | --- | --- |
| **Distance metric** | **k=1** | **k=5** | **k=11** |
| *Euclidean* | 5.04 | 4.96 | 4.38 |
| *MinkowskiL1* | 4.95 | 4.72 | 4.42 |
| *MinkowskiL2* | 5.04 | 4.96 | 4.38 |
| *MinkowskiL3* | 5.58 | 5.35 | 4.58 |
| *MinkowskiL4* | 6.25 | 5.65 | 4.82 |
| *MinkowskiL5* | 6.62 | 5.86 | 5.01 |
| *City Block* | 4.95 | 4.72 | 4.42 |
| *Chebyshev* | 7.67 | 7.05 | 6.42 |
| *Sorensen* | 26.65 | 24.98 | 24.31 |
| *Gower* | 4.95 | 4.72 | 4.42 |
| *Soergel* | 4.99 | 4.72 | 4.43 |
| *Kulczynski d* | 4.99 | 4.72 | 4.43 |
| *Lorentzian* | 5.66 | 5.29 | 5.46 |
| *Canberra* | 4.8 | 4.48 | 4.32 |
| *Intersection* | 4.95 | 4.72 | 4.42 |
| *Wave Hedges* | 4.7 | 4.5 | 4.36 |
| *Czekanowski s* | 4.99 | 4.72 | 4.43 |
| *Czekanowski d* | 51.34 | 50.53 | 49.12 |
| *Motyka s* | 4.99 | 4.72 | 4.43 |
| *Motyka d* | 51.34 | 50.53 | 49.12 |
| *Kulczynski s* | 4.99 | 4.72 | 4.43 |
| *Ruzicka* | 4.99 | 4.72 | 4.43 |
| *Tanimoto* | 4.99 | 4.72 | 4.43 |
| *Inner Product* | 40.68 | 40.68 | 40.09 |
| *Harmonic* | 57.31 | 55.76 | 53.63 |
| *Cosine* | 5.06 | 4.98 | 4.47 |
| *Kumar-Hassebrook* | 5.1 | 4.98 | 4.39 |
| *Jaccard s* | 5.1 | 4.98 | 4.39 |
| *Jaccard d* | 46.01 | 45.74 | 44.01 |
| *Dice s* | 5.1 | 4.98 | 4.39 |
| *Dice d* | 46.01 | 45.74 | 44.01 |
| *Fidelity* | 40.68 | 40.68 | 38.41 |
| *Bhattacharrya* | 57.31 | 55.86 | 53.67 |
| *Hellinger* | 5.01 | 5.01 | 4.27 |
| *Matusita* | 5.01 | 5.01 | 4.27 |
| *Squared-chord* | 5.01 | 5.01 | 4.27 |
| *Squared Euclidean* | 5.04 | 4.96 | 4.38 |
| *Pearson* | 5.11 | 4.97 | 4.49 |
| *Neyman* | 4.87 | 4.78 | 4.28 |
| *Squared* | 5.01 | 4.99 | 4.29 |
| *Probabilistic Symmmetric* | 5.01 | 4.99 | 4.29 |
| *Divergence* | 4.77 | 4.74 | 4.22 |
| *Clark* | 4.77 | 4.74 | 4.22 |
| *Additive symmetric* | 4.99 | 4.99 | 4.31 |
| *Kullbac Leiber* | 57.31 | 55.77 | 53.63 |
| *Jeffreys* | 5.0 | 5.01 | 4.29 |
| *K Divergence* | 57.31 | 55.86 | 53.67 |
| *Topsoe* | 5.01 | 5.01 | 4.28 |
| *Jensen-Shannon* | 5.01 | 5.01 | 4.28 |
| *Jensen difference* | 5.01 | 5.01 | 4.28 |
| *Taneja* | 5.0 | 5.01 | 4.29 |
| *Kumar Johnson* | 4.98 | 5.01 | 4.32 |
| *AvgL* | 4.97 | 4.65 | 4.4 |
| *Vicis-Wave Hedges* | 4.76 | 4.45 | 4.27 |
| *Vicis Symmetric 1* | 4.9 | 4.94 | 4.37 |
| *Vicis Symmetric 2* | 4.91 | 4.96 | 4.29 |
| *Vicis Symmetric 3* | 5.02 | 4.95 | 4.27 |
| *Min-Symmetric* | 4.98 | 4.9 | 4.42 |
| *Max-Symmetric* | 5.02 | 4.98 | 4.31 |
| **Mean value** | **13.18** | **12.88** | **12.15** |

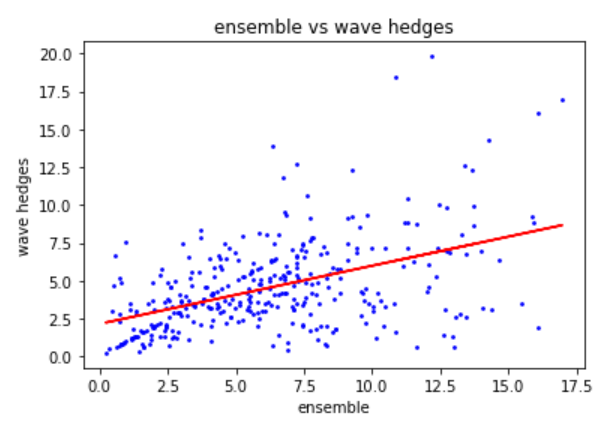
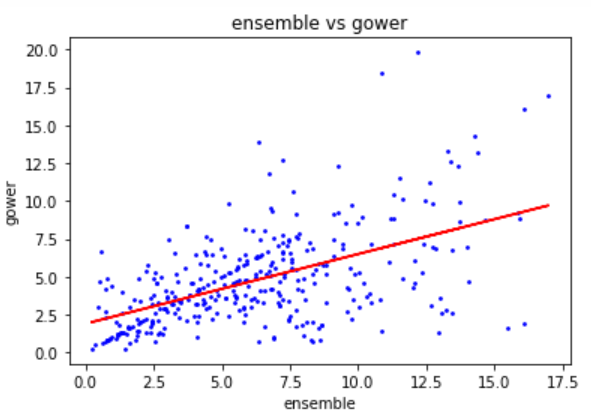
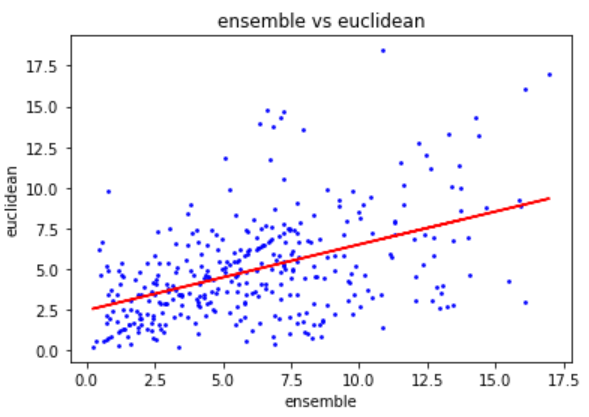
From the results above, we can observe the lowest average error is when k=11. At this point I selected 3 of the best distance metrics to compare the results using an ECDF plot:



In ECDF plots, the most vertical line indicates the best results since the error distance is on the x axis. We can observe in all three plots that k=11 is the best option in this dataset. It is important to notice that the density of points of each dataset affects the election of the k, for dense datasets we will obtain better results with high k values.

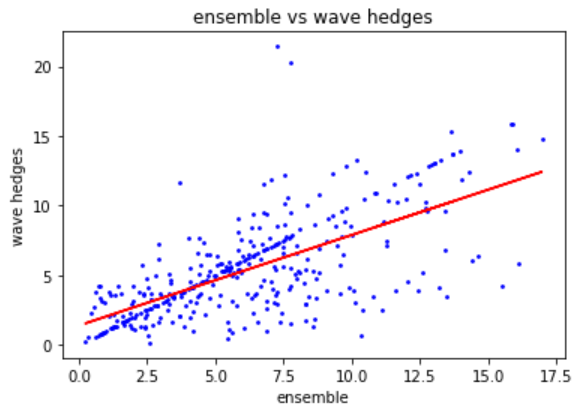
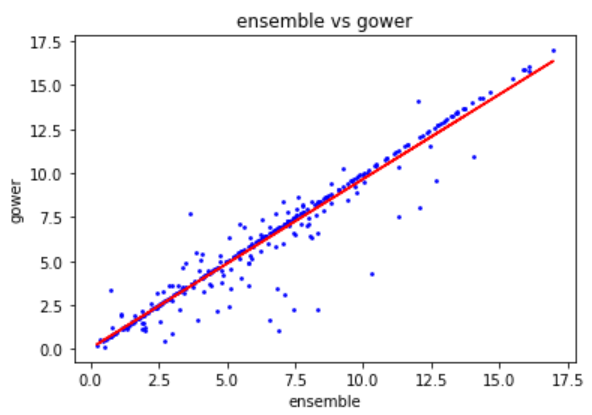
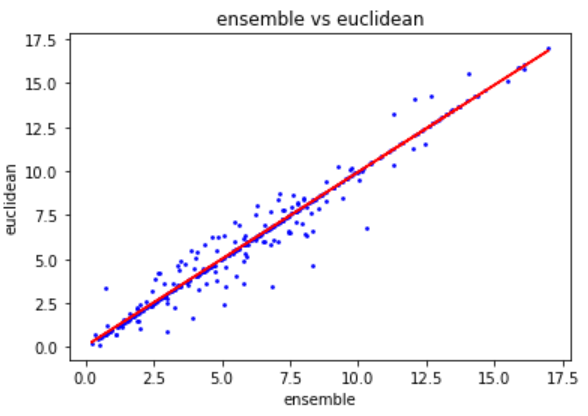
**Ensemble**

The next step to improve the average error is to make an ensemble of the three previous distance metrics and evaluate the results. The following scatter plots show the comparison between the positioning error of the ensemble and the positioning error of the metric alone:



If the points on the scatter plot are very close to the regression line it means the error on both methods is very similar but, in this case, we can see a remarkable dispersion. The points over the line indicate a better performance of the ensemble and the ones under the line indicate better performance of the single metric.

I repeated the same test but instead of applying absolute value, I applied the min-max normalization. The results obtained are slightly different:



In the first two cases, the data points are closer to the line which means that the average error of the euclidean and gower distance is very similar to the ensemble.

# 6. Conclusions and future works

The initial objective of this work was to implement an algorithm based on the k-NN to perform geolocation in indoor environments minimizing the positioning error. After all the tests explained in the previous episodes, the main conclusions would be that it is complicated to find “the best” distance metric or the best configuration to perform a k-NN execution. There are many influential variables to consider:

* The difference between the datasets plays a key role on this problem: density, size, etc.
* Jd

One of the difficulties encountered during the development of the project was the long execution times of the algorithm for some datasets, particularly MAN1 and SIM001. Initially, the idea was to train the model with seven different datasets to compare results but that was finally discarded because of the lack of time. MAN1 and SIM001 included too many fingerprints and the execution of all distance metrics could last up to two or three hours.

Aquest capítol ha d’incloure:

* Una descripció de les conclusions del treball: Quines lliçons s’han aprés del treball?.
* Una reflexió crítica sobre l’assoliment dels objectius plantejats inicialment: Hem assolit tots els objectius? Si la resposta és negativa, per quin motiu?
* Una anàlisi crítica del seguiment de la planificació i metodologia al llarg del producte: S’ha seguit la planificació? La metodologia prevista ha estat prou adequada? Ha calgut introduir canvis per garantir l’èxit del treball? Per què?
* Les línies de treball futur que no s’han pogut explorar en aquest treball i han quedat pendents.

# 8. Bibliography

Llista numerada de les referències bibliogràfiques utilitzades dins la memòria. A cada lloc on s’utilitzi una referència dins el text, cal indicar-la citant el número de la referència, per exemple: [7].

És molt important incloure **totes** les referències utilitzades i citar-les apropiadament, és a dir, incloent tota la informació necessària per identificar la referència. La informació mínima que cal incloure segons el tipus de referència és:

* **Llibre:** Autors, Títol, Edició (si s’escau) Editorial, Ciutat, Any.
* **Article de revista:** Autors, Títol, Nom de la Revista, Número de Pàgina inicial i final, Número de la revista / Volum, Any.
* **Web:** URL i data en que s’ha visitat.

# 9. Annex

Llistat d’apartats que són massa extensos per incloure dins la memòria i tenen un caràcter autocontingut (per exemple, manuals d’usuari, manuals d’instal·lació, etc.)

Depenent del tipus de treball, és possible que no calgui afegir cap annex.