Indoor robot positioning system based on Wi-Fi signals: An implementation using machine learning.

Apostolos Apostolou, Mechanical Engineering Undergraduate Student at University of Cyprus

August 2023

Introduction

Defining the position of a mobile robot (MR) is essential for enabling autonomous navigation. Various methods exist for detecting MR position, including Inertial Measurement Units (IMU), Computer Vision, Bluetooth, and others [1]. This project focuses on a straightforward implementation of a method to determine MR position using Wi-Fi signals and machine learning. While this project doesn't delve into robot navigation, it concentrates solely on location detection.

The system

The central idea of this system is to determine the MR's position by assessing Wi-Fi signal strength at different points within an indoor environment. By measuring the Wi-Fi signal strength from various SSIDs (3-4 Wi-Fi repeaters at different positions within the building) and associating these signals with specific locations on a 2D Cartesian coordinate system, we can predict the MR's location. This prediction is achieved through a regression model, which takes the Wi-Fi signal strengths for all SSIDs at the new location as input and outputs the predicted MR location. Wi-Fi signal strength is typically expressed in dBm, instead of mW, to avoid unwieldy decimal numbers [2].

The following diagram illustrates how the machine learning algorithm predicts the MR's location based on Wi-Fi signal strength. The system scans and detects the three signals at the robot's location (marked by red star). A regression algorithm is then applied to predict the location using known location-signal pairs. Each square around the robot’s location indicates the 3 known signals at those locations. Each color at the square represents the signal from the router of the same color.

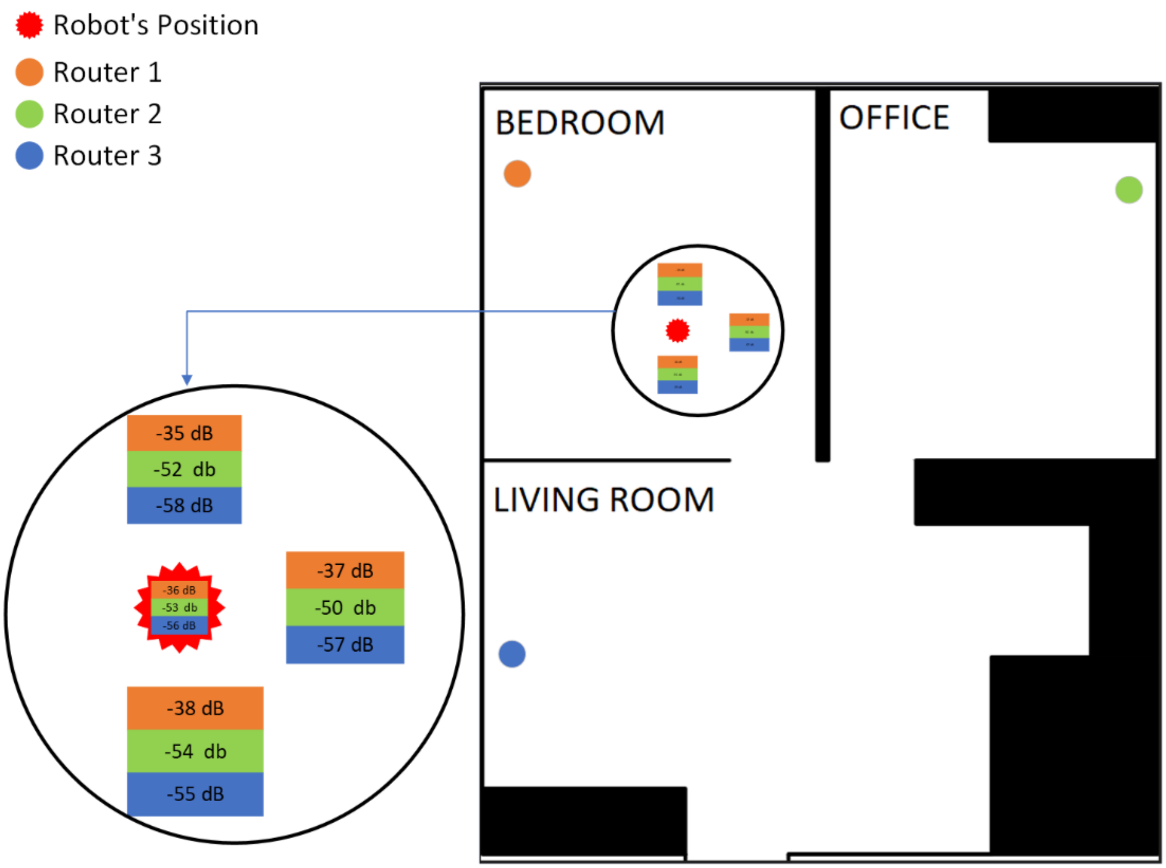

Figure 1: Graphical representation of how the system reads the Wi-Fi signal's strength and predicts the robot's location.

The coordinates of the routers, as shown above and as used for testing in the project are the following (X, Y):

Router 1: (10, 620)

Router 2: (550, 600)

Router 3: (50, 200)

Implementation

For this Indoor Positioning System (IPS) to function, manual mapping of the environment is necessary. This map serves as the backdrop for plotting the live location.

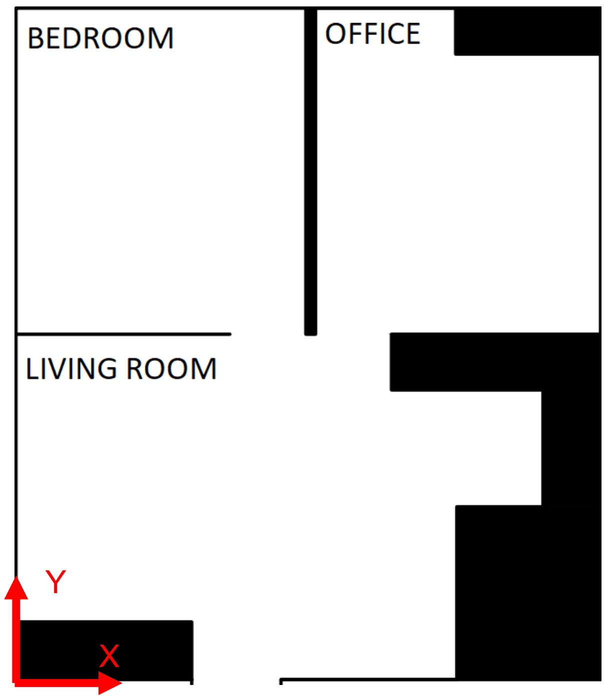

Figure 2: Floor map of the apartment that is used for testing in the project.

It's also crucial to measure three different Wi-Fi signals at various (X, Y) coordinates within the building. In the example image, the bottom-left corner corresponds to (0, 0), and the top-right corner corresponds to (609, 700) with numbers representing centimeters. A Python script was used to collect measurements at 83 points, recording results (X, Y, Signal 1, Signal 2, Signal 3) in an Excel file.

Figure 3: Coordinates marking at the apartment's floor during system implementation.

Figure 4: Computer used as mobile robot during the tests. In real robot the computer can be a Raspberry Pi for example.

The algorithm

To predict the MR's location in real-time as it moves through space, the model must undergo training. Data collection is achieved through the aforementioned Python script, gathering Wi-Fi signal strengths for three different SSIDs across numerous building locations. The algorithm's input, denoted as X', is an array with three numbers (representing the Wi-Fi signal strengths in dBm), while the output Y' is an array with two numbers (representing X and Y coordinates). The dataset was randomly shuffled and divided into training (80%) and testing (20%) sets.

The chosen training models fulfilled two primary criteria. Firstly, the model needed to be a regression model, projecting a continuous value [3]. Secondly, due to the logarithmic nature of dBm measurements [2], a non-linear model was preferred. Training efficiency was also considered. Two final algorithmic options were explored: K-Nearest Neighbor Regression (KNN) and Random Forest Regression (RFR).

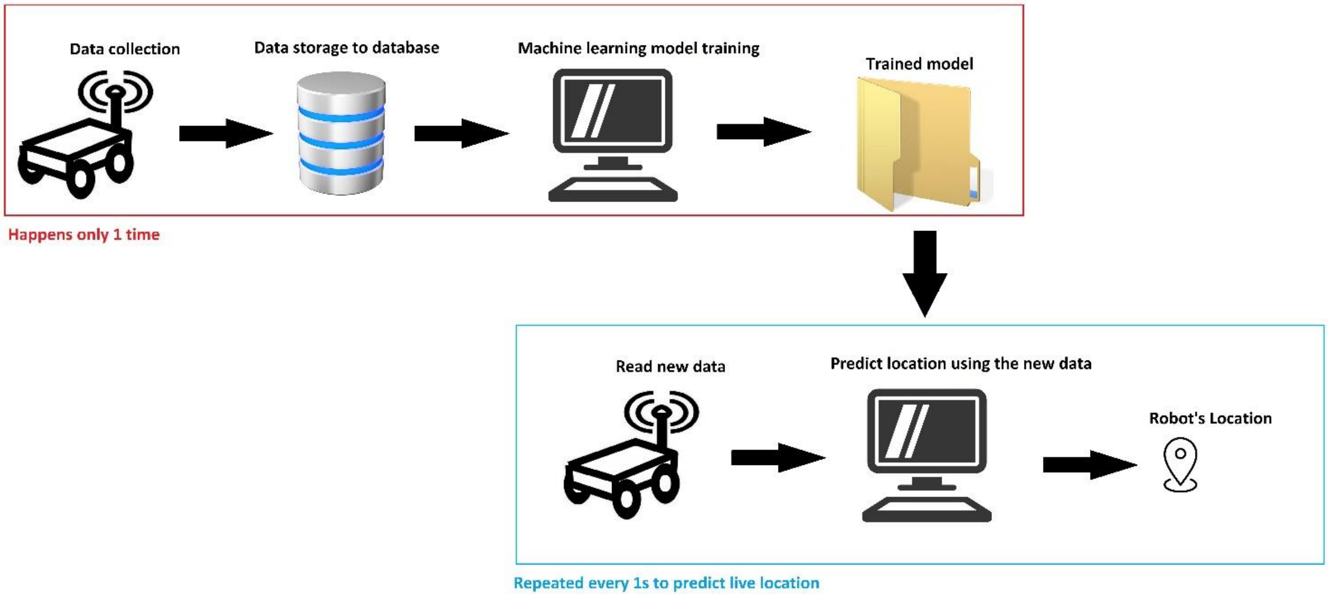

Figure 5: System overview.

Below are the results of the two algorithms, along with Figures 5 and 6 illustrating predicted vs. actual locations for evaluating model accuracy.

The accuracy of each algorithm is measured using an average distance error (ADE), measured in meters, for each model, which gives an estimation of the error between the actual location of the robot in relation to the predicted location (using the Euclidean distance [4] between the 2 points).

$$ADE = \sum_{i=1}^{n} \frac{\sqrt{(Xi - \widehat{Xi})^2 + (Yi - \widehat{Yi})^2}}{n}$$

Where: i: i_{th} data point in testing set, n: total number of data points in testing set, Xi: prediction of X coordinate, $\widehat{Xi}$: actual value of X coordinate, Yi: prediction of Y coordinate, $\widehat{Yi}$: actual value of Y coordinate.

K - Nearest Neighbor Regression Algorithm (KNN):

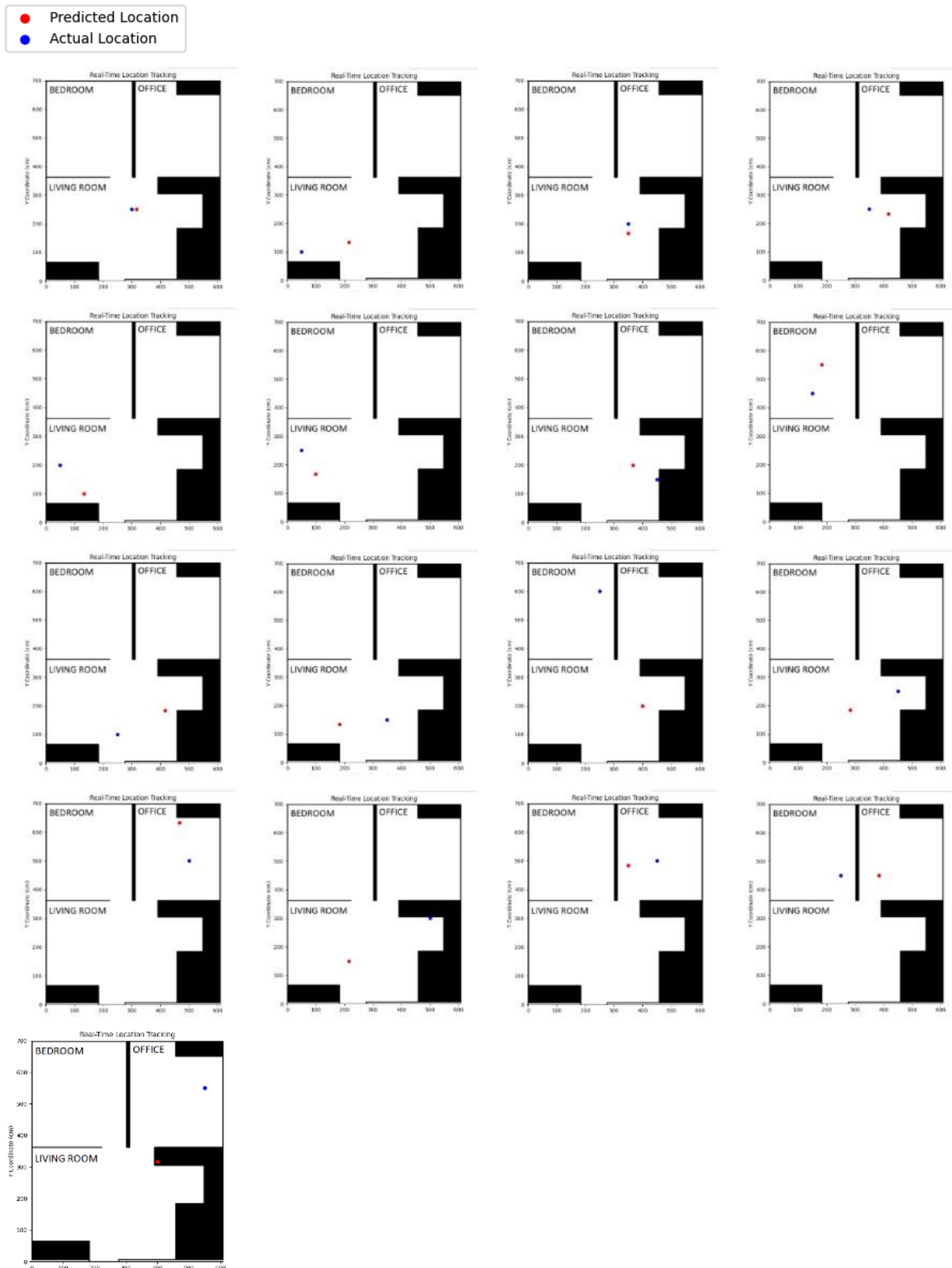

Figure 6: KNN - Predicted vs Actual Location on testing data.

ADE: 1.56 meters

Accuracy of predicting the correct room (living room, bedroom, or office): 82.35 %

Random Forest Regression Algorithm (RFR):

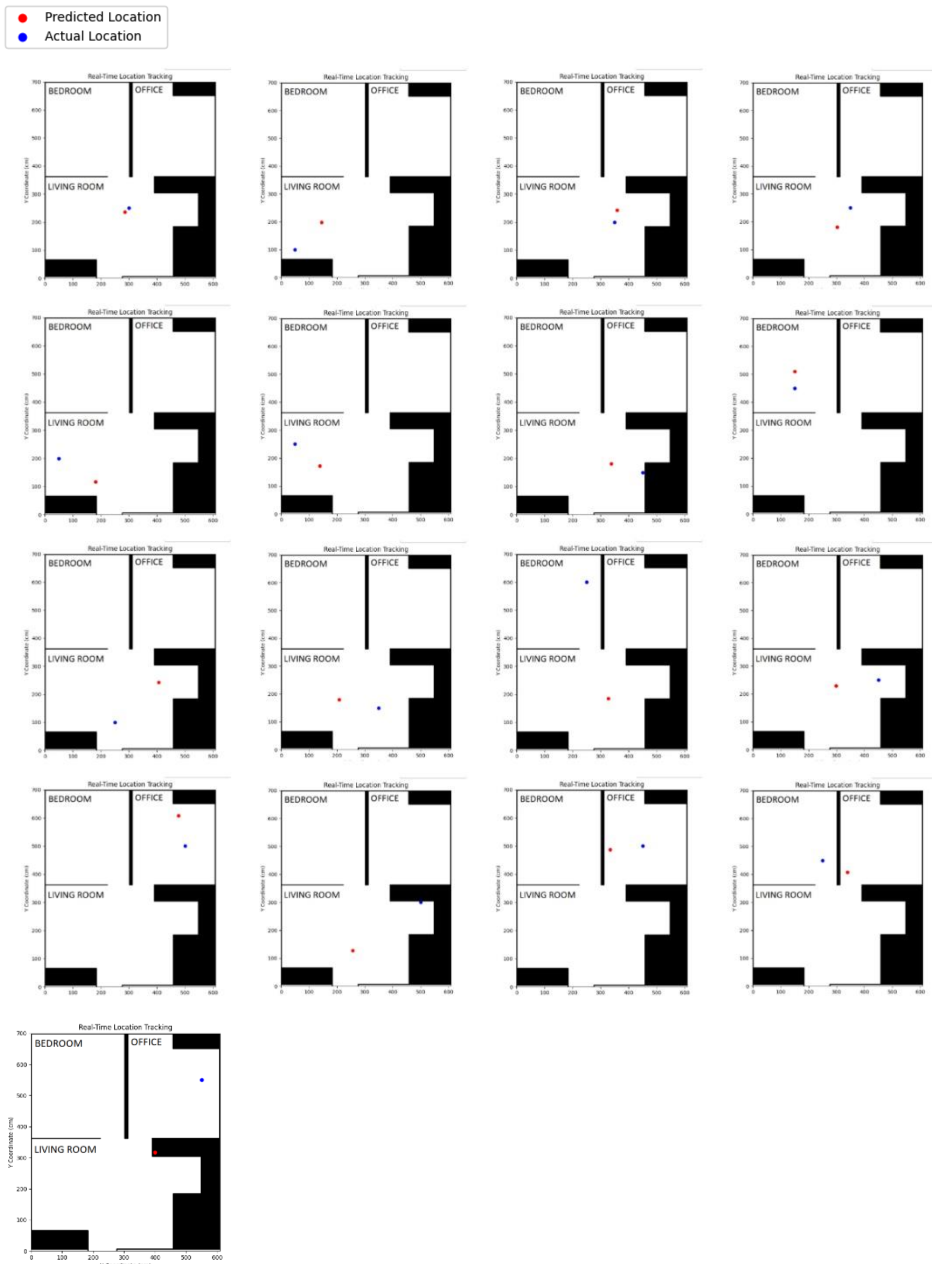

Figure 7: RFR - Predicted vs Actual Location on testing data.

ADE: 1.45 meters

Accuracy of predicting the correct room (living room, bedroom, or office): 88.24%

Overall, the RFR seems to work better as it has a lower ADE and better accuracy at predicting the corrected room in which the robot is located.

Discussion

This approach, although is not that accurate, offers a cost-effective method to estimate an MR's location. However, improvements are feasible. One enhancement involves using external Wi-Fi antennas instead of internal ones (e.g., Raspberry Pi’s internal antenna). This change facilitates universal usage across multiple MRs without re-measuring. More data points can be collected to enhance prediction accuracy. Exploring the implementation with additional Wi-Fi repeaters (4-5 instead of 3) is also worth considering. Notably, while this system provides a rough idea of the MR's position, manual floor marking, and signal measurements make it challenging for large, crowded spaces.

References

1) Huang J, Junginger S, Liu H, Thurow K. (2023) Indoor Positioning Systems of Mobile Robots: A Review. *Robotics*.
2) Pawar, Yogesh & Prabhu, Tejas & Naik, Vikrant & Shelke, Saurabh. (2017). Wi-Fi Signal Strength and Analysis
3) Géron, A. (2019). *Hands-on machine learning with Scikit-Learn, Keras and TensorFlow: concepts, tools, and techniques to build intelligent systems* (2nd ed.). O’Reilly.
4) Britannica, T. Editors of Encyclopaedia *Euclidean distance*. *Encyclopedia Britannica*. https://www.britannica.com/science/Euclidean-distance