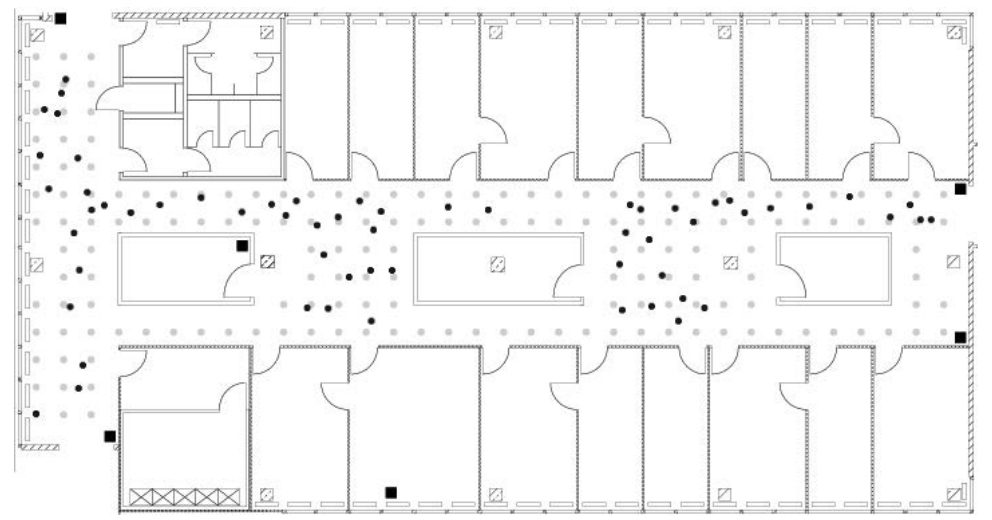
**Abstract**

**Introduction**

With the advent of Wi-Fi and local area networks, devices like Real-Time Location Systems (RTLS) can be leveraged for location positioning of an object in a specified area in real time. This is made possible by way of a continuous communication and feedback between a device held by the object being tracked and the beacons or receiver by the host. Examples of RTLS’s include Infrared, Bluetooth, Cellular, and Radio Frequency Identification (RFID). To operate RTLS, a scanning device is required to locate an object (such as a cell phone or laptop) based on the angle and coordinates of the object being tracked. You can use multiple scanning devices to triangulate the exact location of the object using the combination of angles in which the signal is triggering the respective receivers. Utilizing a series of wireless network signals in an office building, we will be able to detect the exact location of various objects in real time using different combinations of mac devices. Here, we can perform an unweighted and weighted k-nearest neighbors (k-NN) analysis to predict the location of the Online data using the offline data. To go further, we will be seeing if we can better predict the location of the online data using different combinations of the mac addresses available.

**Data**

For this project, we will be leveraging two separate datasets for our analysis. One of which is a reference set named “offline” which contains signal strength measurements from a hand-held device on a gridwork of 166 different points, all of which were spaced 1 meter apart. This gridwork is located in the hallways of a one floor building at the University of Mannheim. The other dataset is titled “online” which we will be using for testing our k-NN model to predict the location. This dataset includes 60 different locations chosen at random with 110 signals measured from them across each point. In figure 1.1 below, you can see a map of the “online” test locations (black dots) overlaid with the “offline” training locations (grey dots). Both datasets contain the same features and will require the same procedures for cleaning.



*Figure 1.1. The floor plan of our experimentation environment. This makes up a 1-floor building in the University of Mannheim with the offline (grey dots) points placed about a meter apart from one another throughout the building. In addition, you can see the placement of the online points scattered throughout for testing.*

|  |  |  |  |
| --- | --- | --- | --- |
| *Feature Name* | *Description* | *Original Data Type* | *Converted Data Type* |
| Time | Timestamp in milliseconds since midnight, Jan 1. 1970 | Character | Numeric |
| id | MAC address of the scanning device | Character | Character |
| pos | The physical coordinate of the scanning device which can be broken out in a latitudinal or longitudinal coordinate plane. | Character | Numeric |
| degree | Orientation of the user carrying the scanning device in degrees | Character | Numeric |
| MAC | MAC address of a responding peer (e.g., an access point or a device in adhoc mode) with the corresponding values for signal strength in dBm (decible milliwatts), the channel frequency and its mode (access point = 3, device in adhoc mode = 1 | Character | Character |

*Table 1.1. The raw data collected from* [*http://rdatasciencecases.org/Data/offline.final.trace.txt*](http://rdatasciencecases.org/Data/offline.final.trace.txt) *can be found here in it’s original form. It includes rudimentary details on the time in which the record was taken, the id number of the scanning device, the position of the reading, the angle of reading and the Mac Address of the responding peer device. This data will need to be cleaned in order for us to further analyze.*

The first cleaning method we will employ will be to break up the position variable into separate variables which we can use to triangulate the location. In our raw dataset, we have position values for latitude, longitude and elevation separated by commas, which we will convert into PosX, PosY and PosZ. Upon further cleaning, we were able to determine that there was only one unique value for PosZ at 0 (which made sense considering the experiment took place in a one story building), we had the liberty to drop the variable. Additionally, running a procedure to check on the number of unique variables in the ScanMac column yielded only a single unique value, so we can drop that one as well.

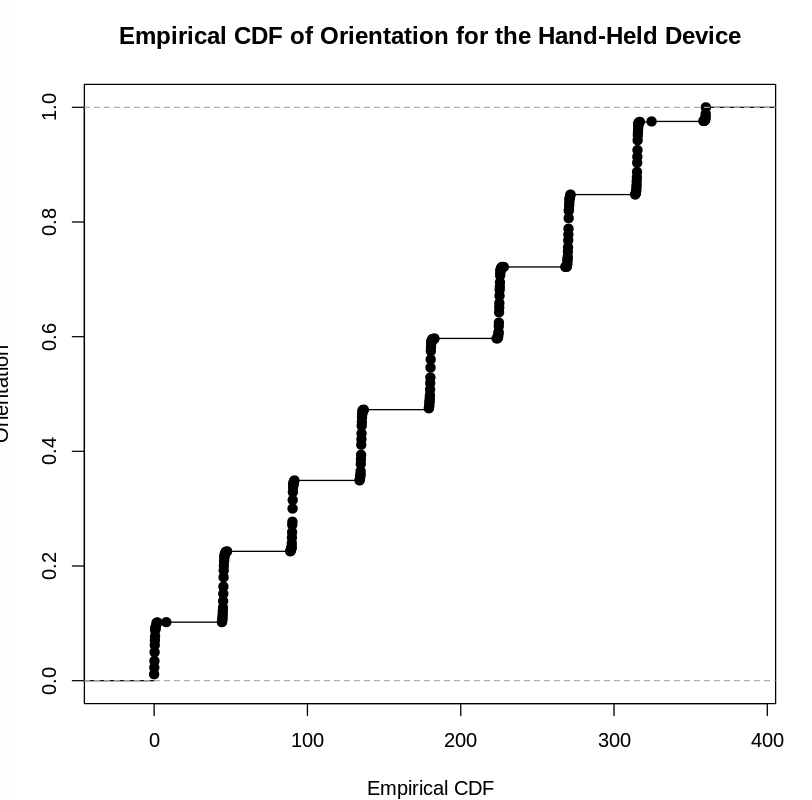
In our documentation, we found that our type of device was mixed between the values 1 and 3, which we may want to clarify more. Reviewing our documentation, we will only want to focus on fixed access points as that is more relevant to our study of predicting device locations using a fixed set of receivers. So, moving further, we will remove the adhoc instances in our dataset.

The Time measurement is something that we will want to make an adjustment to so that we can more easily analyze in the future. As mentioned prior, the time data is based on the number of milliseconds from a specific date (which could possibly be arbitrary), so we can change to a Year-Month-Day-Time format. But first, we can divide the number of milliseconds to seconds. This leaves us with the following features that we will use across our offline and online data. Additionally, we will remove the channel feature since it is strictly a character code that contains redundant identifiers of Mac Address, signal strength, frequency and mode that may play an unfair role in our predictive modeling.

|  |  |  |
| --- | --- | --- |
| Variable | Description | Datatype |
| Time | Year-Month-Day-Hours-Minute-Seconds | POSIXt |
| posX | X coordinate of the device | Numeric |
| posY | Y coordinate of the device | Numeric |
| Orientation | Orientation angle of the device (in degrees) | Numeric |
| Mac Address | The device address of the Mac reciever | Character |
| Signal | Signal Strength of the device (in dBM) | Numeric |
| Raw Time | The raw timestamp / 1000 | Numeric |

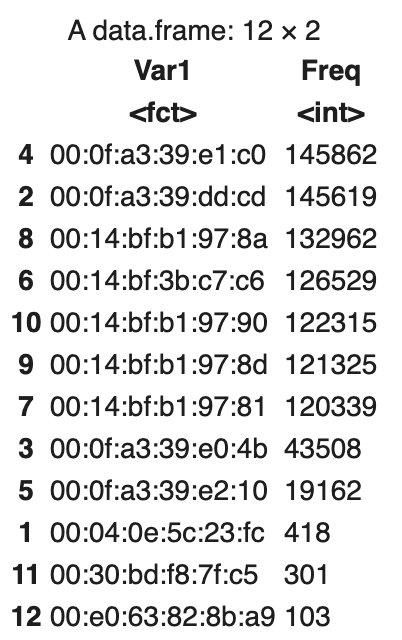
*Table 2.1 Our final data table following our cleaning procedure. Includes the dropping of our PosZ and scanMac value as well as our channel value.*

Next, we will look at the orientation column of our dataset, we can see that we have a wide variety of angles available in clusters around the expected angles (such as 179 or 181) as shown in figure 2.1. Since we are going to focus on measure signal strength at 8 orientations in 45-degree increments, we will round each of our orientations to the nearest 45 degree increment. Additionally, we will try to map values close to 360 so that they line up back to zero.



*Figure 2.1. The location of the orientation values as it relates to the empirical cdf. We can see that at each major orientation (such as 45, 90, 135, 180, ect) we are scattered around these values. So our cleaning procedure is going to round these to the nearest 45 degree angle.*

The last cleaning procedure we will follow is to review the frequency of the mac addresses in place for our models. Since the documentation mentioned that the mac addresses ending in ‘c5’, ‘a9’, ‘fc’, ‘10’, and ‘4b’ were either not on the correct floor, or were not turned on the whole time, we will want to drop these addresses so that we are only working with the remaining 7 addresses that have a frequency of over 120,000, and dropping the ones with less, as per figure 3.1.



*Figure 2.1. Frequency count of Mac addresses in our study. We will be dropping the addresses that have a frequency count below 120,000.*