**WiFi Locationing Analysis Report**

**FOR**

**Kathy Narim-Nomstrom**

**BY**

**Clark Humphrey**

**Introduction:**

A client is developing a system to be deployed on large industrial campuses, in shopping malls, et cetera to help people to navigate a complex, unfamiliar interior space without getting lost. I investigated the feasibility of using "wifi fingerprinting" to determine a person's location in indoor spaces. Wifi fingerprinting uses the signals from multiple wifi hotspots within the building to determine location, analogous to how GPS uses satellite signals.

Research was conducted and data analysis performed to evaluate multiple machine learning models to see which produces the best result, enabling us to make a recommendation to the client. The task was to utilize Python and evaluate at least 3 different algorithms for predicting location.

**Data Sets Analyzed:**

The data set analyzed is a large database of wifi fingerprints for a multi-building industrial campus with a location (building, floor, and location ID) associated with each fingerprint.

The data set and it’s details are at: <http://archive.ics.uci.edu/ml/datasets/UJIIndoorLoc>.

A paper containing a detailed overview of the dataset and how it was collected is at: [UJIIndoorLoc - A New Multi-building and Multi-floor Database for WLAN Fingerprint-based Indoor Localization Problems](UJIIndoorLoc%20-%20A%20New%20Multi-building%20and%20Multi-floor%20Database%20for%20WLAN%20Fingerprint-based%20Indoor%20Localization%20Problems)

Or at: [UJIIndoorLoc - A New Multi-building and Multi-floor Database for WLAN Fingerprint-based Indoor Localization Problems](https://s3.amazonaws.com/gbstool/courses/614/docs/UJIIndoorLoc%20-%20A%20New%20Multi-building%20and%20Multi-floor%20Database%20for%20WLAN%20Fingerprint-based%20Indoor%20Localization%20Problems.pdf?AWSAccessKeyId=AKIAJBIZLMJQ2O6DKIAA&Expires=1578819600&Signature=jZRoVjxu3VX87%2BCFRMIm3Jnuj%2Bk%3D)

The data set covers three buildings of Universitat Jaume I with 4 or more floors and almost 110,000m2. The data is provided in 2 sets: a Training Data set of 19,937 records and a Validation Data set of 1,111 records.

Each data record contains 529 elements that make up the wifi fingerprint. Each wifi fingerprint is characterized by the detected Wireless Access Points (WAPs) and the corresponding Received Signal Strength Intensity (RSSI). The RSSI values are represented as negative integer values ranging -104dBm (extremely poor signal) to 0dbm. The positive value 100 is used to denote when a WAP was not detected. Each record contains the location’s fingerprint of RSSI values from 520 different WAPs. The 9 additional data elements of each record are:

* Attribute 521 (Longitude): Longitude. Negative real values from -7695.9387549299299000 to -7299.786516730871000
* Attribute 522 (Latitude): Latitude. Positive real values from 4864745.7450159714 to 4865017.3646842018.
* Attribute 523 (Floor): Altitude in floors inside the building. Integer values from 0 to 4.
* Attribute 524 (BuildingID): ID to identify the building. Measures were taken in three different buildings. Categorical integer values from 0 to 2.
* Attribute 525 (SpaceID): Internal ID number to identify the Space (office, corridor, classroom) where the capture was taken. Categorical integer values.
* Attribute 526 (RelativePosition): Relative position with respect to the Space (1 - Inside, 2 - Outside in Front of the door). Categorical integer values.
* Attribute 527 (UserID): User identifier (see below). Categorical integer values.
* Attribute 528 (PhoneID): Android device identifier (see below). Categorical integer values.
* Attribute 529 (Timestamp): UNIX Time when the capture was taken. Integer value.

The Validation Data set contains the value 0 for the following data elements:

* SpaceID
* RelativePosition
* UserID

Exploration of the data revealed the following:

* Floors:
  + Buildings 0 and 1 have 4 floors
  + Building 2 has 5 floors
  + The distribution of records per building per floor in the Training Data set are:

A picture containing drawing

Description automatically generated

* Space IDs:
  + Space ID is comprised of 123 levels
  + The range of Space IDs by building is contained in the figure below:

A screenshot of a video game

Description automatically generated

* Relative Position:
  + The distribution of records with Relative Position = 1 or 2 is shown below:

A picture containing drawing

Description automatically generated

* WAP values
  + Analysis revealed that 55 of the 520 WAPs (10.6% of the location features) had values of “100” for all records. Since they have the same value for all locations, this indicates they likely don’t contain useful information for locating any particular location. Therefore, they’re likely not needed for training algorithms to determine location.

**Data Preparation for Analysis:**

The data sets did not require cleaning to prepare for analysis. Neither set was missing data.

I chose to create a single location field for the models to predict using a classification method. The single field was created by concatenating Floor, BuildingID, SpaceID, and RelativePosition with a dash (“-”) separator after converting each to a “str” type. The result was a single field named “LOCATION” comprised of “BUILDINGID-FLOOR-SPACEID-RELATIVEPOSITION” such as "1-2-106-2", "1-2-106-2", "1-2-103-2", "1-2-102-2" ...

This created 905 levels or unique locations. *Therefore, the task became to create a model or models to predict which one of these 905 locations the user is positioned.*

Therefore, the following data fields were deemed unneeded for the analysis and removed from the dataset to avoid any issues with the model development: Longitude, Latitude, UserID, PhoneID, Timestamp.

The records in the resulting data set contained 521 data elements. 1-520 WAP RSSI values and 1 data element of “LOCATION”. This is referred to as the “Full Data Set”.

Another data set was created by 1) removing the 55 WAPs that had only “100” values for all records, and 2) removing all records with RELATIVEPOSITION = 1 (inside the space). This is referred to as the “Reduced Data Set”. This reduced data set was created to compare algorithmic performance and the validate the assumption the 55 WAPs contained no meaning information to any particular location, and that the user would be satisfied navigating to the outside of the destination space (e.g. the hallway door of the room).

The Validation Data set was NOT used in this analysis. Since it did not contain valid values for SpaceID and RelativePosition, it could not be used to train or test a model.

In order to evaluate models by building, the “Full Data Set” was used to create 3 additional subdivided data sets - one for each building - resulting in the following:

* Building 0 data set: 5249 observations; Location field contained 259 levels (e.g. “Floor\_SpaceID\_RelativePosition”)
* Building 1 data set: 5196 observations; Location field contained 243 levels (e.g. “Floor\_SpaceID\_RelativePosition”)
* Building 2 data set: 9492 observations; Location field contained 403 levels (e.g. “Floor\_SpaceID\_RelativePosition”)

In order to evaluate models that can predict building only and building and floor only, the following data sets were created from the “Full Data Set”:

* Building only: 19,937 records; Location field contained 3 levels (e.g. “BuildingID”)
* Buiding/Floor only: 19,937 records; Location field contained 13 levels (e.g. “BuildingID\_Floor”)

In order to evaluate models that can predict location when building and floor are known, the following data sets were created from the Building 1 data set:

* Building 0 Floor 0: 1059 observations; Location field contained 54 levels (e.g. “SpaceID\_RelativePosition”)
* Building 0 Floor 1: 1356 observations; Location field contained 67 levels (e.g. “SpaceID\_RelativePosition”)
* Building 0 Floor 2: 1443 observations; Location field contained 70 levels (e.g. “SpaceID\_RelativePosition”)
* Building 0 Floor 3: 1391 observations; Location field contained 68 levels (e.g. “SpaceID\_RelativePosition”)

Similar data sets were also created for each floor of Buildings 1 and 2.

**Selecting Classification Algorithms:**

**Predicting Location with the full data set:**

The initial modeling attempt was to use the “Full Data Set” to see how well the algorithms would predict LOCATION with such a large data set. The training set was a random selection of 80% of the “Full Data Set”. The “out of the box” algorithm accuracy results are shown below:

MEAN ACCURACY:

* RF: 0.795347
* CART: 0.582795
* KNN: 0.593266
* SVM: 0.595084

A screenshot of a cell phone

Description automatically generated

**Predicting Location with the reduced data set:**

The next modeling attempt was to use the “Reduced Data Set” to see if algorithms would better predict LOCATION with a smaller data set. The training set was a random selection of 80% of the “Reduced Data Set”. The “out of the box” algorithm accuracy results are shown below:

MEAN ACCURACY:

* RF: 0.797003
* CART: 0.584600
* KNN: 0.598828
* SVM: 0.604698

A screenshot of a cell phone

Description automatically generated

CONCLUSION: The results are basically identical to the “Full Data Set”. So, no advantage to using the “Reduced Data Set”. The “Full Data Set” was used for all further analysis just in case the additional information it contained was meaningful to the algorithms.

**Predicting the Building:**

In order to try to improve the algorithm performance, I decided that a reduction in levels of the Location field would likely improve the results – fewer levels to decide from should improve the accuracy. The first attempt was to narrow the Location field to a single building (data set comprised of 3 factors) and find the best algorithm to predict the building the user is in. The data set used was the “Building Only” data set described above. The training set was a random selection of 80% of the “full data set”. The “out of the box” algorithm accuracy results are shown below:

MEAN ACCURACY:

* RF: 0.997617
* CART: 0.995925
* KNN: 0.997617
* SVM: 0.997931

CONCLUSION: Ability to predicting the correct building is practically 100% for any of these models.

**Predicting Location after Building is Known:**

Now that we have a highly accurate model for determining the building the user is in, I decided to evaluate algorithms to predict the location within each building. In doing so, the Building 0, Building 1, and Building 2 data sets were used (described above). The training sets were a random selection of 80% of each data set respectively. The “out of the box” algorithm results were:

BUILDING 0:

MEAN ACCURACY:

* RF: 0.746843
* CART: 0.566801
* KNN: 0.532521
* SVM: 0.536551

A screenshot of a cell phone

Description automatically generated

BUILDING 1:

MEAN ACCURACY:

* RF: 0.849371
* CART: 0.681420
* KNN: 0.631613
* SVM: 0.602499

A picture containing clock

Description automatically generated

BUILDING 2:

MEAN ACCURACY:

* RF: 0.800341 (0.016440)
* CART: 0.557219 (0.015852)
* KNN: 0.602131 (0.016327)
* SVM: 0.628466 (0.025162)

A screenshot of a cell phone

Description automatically generated

The Random Forest (RF) algorithm is the best for each building.

The resulting accuracy of these models range from 0.746843 to 0.849371, depending on the building. This “find the building first” approach is an improvement over the 0.795347 accuracy from the approach to predict location straight from the full-data set, except for Building 0.

**Predicting the Building and Floor:**

Next, I evaluated lowering the number of factors again to improve algorithm performance by trying to predict building and floor first, then location on that floor.

A similar approach to “building only” was used to determine if building and floor could be accurately predicted. The data set used was the “Building/Floor Only” data set described above (with 13 factors of “building-floor”). The training set was a random selection of 80% of the “full data set”. The “out of the box” algorithm accuracy results are shown below:

MEAN ACCURACY:

* RF: 0.993605
* CART: 0.943069
* KNN: 0.965076
* SVM: 0.972099

CONCLUSION: The RandomForest (RF) algorithm can predict the correct building and floor with a very high level of accuracy.

**Predicting Location after Building\_Floor is Known:**

Now that we have a highly accurate model for determining the building and floor the user is in, I decided to evaluate algorithms to predict the location within the floor of each building. In doing so, the Building 0, Floor 0, 1, 2, 3 data sets were used (described above). The training sets were a random selection of 80% of each data set respectively. The results were:

![A screenshot of a cell phone

Description automatically generated]()

Unfortunately, these results are mixed. Floors 0 and 1 are better than the results when just the Building 0 is known (RF accuracy of 0.746843), however Floors 2 and 3 are worse. Interestingly, the overage of these 4 values is close to the “Building 0 is known” value. Therefore, the approach of finding the location after the Building-Floor is known was not pursued any further.

**Conclusions:**

**Best Approach Found:**

The best approach I found was to use the RandomForest algorithm to predict the Building the user is in (with an accuracy of 0.997617), then use the RandomForest algorithm to determine the location of the user within that building (with an accuracy ranging from 0.746843 to 0.849371 depending on the building).

**Tuning the Algorithms:**

Next, I attempted to improve upon the above results by tuning the RandomForest algorithm parameters for the models that predict the location of the user once the building is known. These models were also evaluated for Accuracy and Kappa against a test data set (instead of the training set) and their confusion matrix was generated and viewed.



The tuned algorithms increased performance for each Building by a few percentage points when compared to the “out of box” algorithm results.

CONCLUSION: These tuned algorithms are the recommended ones to implement and the accuracy expected in practice are those shown in the table above.

**Acceptability:**

Although the accuracies resulting from this approach may appear unacceptable to a user, it still may satisfy the user in the vast majority of instances. For example, if the algorithm successfully navigates the user to a location on the proper floor and adjacent to the location they’re seeking, that should be sufficient in most cases since the user will simply recognize this from nearby office numbers and other markings in the building and easily get to their destination.

A confusion matrix was created from each Building’s test set data to determine how many predicted instances are in these “nearby” locations when the algorithm does not predict the exact location. The matrix for Building 0 is in the attached MS Excel file. The visualization this matrix provides indicates that the “missed” predictions are almost always nearby the actual location. Therefore, this approach is likely to produce acceptable results and should be tested with users.



If the results are not acceptable because the users find themselves on the wrong floor too often, then we should fully develop the models for “Predicting Location after Building-Floor is Known” for each floor and use them instead. The model to predict Building-Floor is extremely accurate (e.g. 0.993605), so getting them to proper floor will work every time. Even if the overall accuracy for these models are a little lower, the user will be on the proper floor and likely be close enough to their destination to easily find it.

**Recommended Further Investigations:**

The following could be performed to determine if better performing algorithms can be derived:

* Perform the same “Predicting Location after Building\_Floor is Known” analysis for Buildings 1 and 2 that was done for Building 0. Then select the best approach to use based on the Building-Floor result. (e.g. if on a Building-Floor with high accuracy results, use it. If not, use the “Building Known” approach.) However, in practice, this may prove to be no better than just using the “Building Known” approach since the places that produce better “Building-Floor” results will likely average out with the poorer places and produce an overall result close to the “Building Known” only approach.
* Perform the “Predicting Location after Building-Floor is Known” analysis but use a regression approach with latitude and longitude instead of the classification approach with “SpaceID-RelativePosition”. Since floor is already known at that point, we only need a “x-y” position which latitude and longitude provide.
* Repeat all of the analysis using the “Reduced Data Set” (without the 55 WAPs and without “RelativePosition=1” data records) to see if it yields better results. Rationale is that the user is likely to be satisfied if they can be navigated to the coordinate outside the office, classroom, etc. they are seeking. This could yield better results since the number of factors will be reduced. However, this is not assured given the outcome of the analysis with the “Reduced Data Set” described early in this report.
* Research suggests that a customized KNN algorithm called “enhanced weighted K-nearest neighbor (EWKNN)” yields better results than KNN for wifi fingerprinting. Therefore, this algorithm should be tried.
* Consider utilizing an open source indoor positioning system as a starting point to any development/deployment that we do for our client. This could significantly shorten our time to deployment. Options include:
  + Redpin: an open source indoor positioning system that was developed with the goal of providing at least room-level accuracy. We can evaluate it’s locationing algorithm and choose to use it or implement ours – whichever is best. <http://redpin.org/>
  + Indoor Location: The first open-source unified framework to manage locations coming from any Indoor Positioning Technology. <https://www.indoorlocation.io/>
  + FIND: Open-source framework for internal navigation and discovery. <https://www.internalpositioning.com/>