# Indoor Localization Implementation Based on Machine Learning Method with Data Dimensionality Reduction Analysis

Reported by: WANG ZEKAI

(UROP 1000, 2022-23 summer, Prof. Gary Chan's Group)

Content Overview:

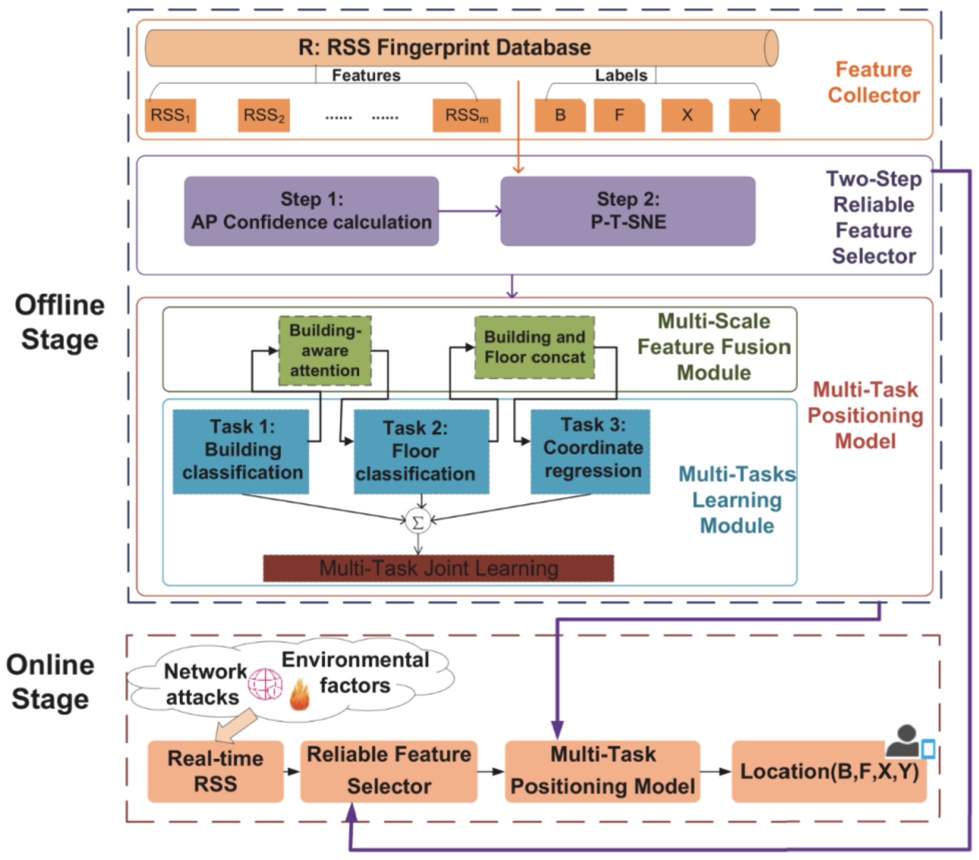
## Abstraction

Indoor localization has been under research since we spend more time using our mobile devices inside huge building complexes with multiple floors. Industry and academia have introduced different approaches, they vary in their input data. For example, some use ultra-wideband (UWB), Bluetooth signal or device built-in accelerometer as the input data for localization [1]. This report will focus on using WIFI fingerprints to estimate the user's location by applying machine learning technique inside an MTR station.

The result of this method is, unfortunately, unsatisfactory from the real application point of view because of the sparse accessing point. However, it is believed that the key to improving the accuracy of our estimation is how to process the input data correctly, especially when the training data is limited in size. Therefore, another focus of this report is to analyze different dimensional reduction / feature extraction algorithms applying onto the input data, compare the strength and weaknesses of different algorithms. And finally review my implementation and raise improvements.

## Current Approach

The main idea is gained from research which introduced a concept called 'two-step reliability' method in handling input data, together with multi-task learning module [2]. They first put RSS fingerprint data into a feature collector, to shape the input into a processable form. Then, pass the input into a 'two-step reliable feature selector', this is to select data with higher reliability and put selected data into a feature selector to extract useful features. The processed data is then fed into the classification model to classify the building and the floor, and a coordinate regressor to estimate the user's location. Fig 1 shows the flow of their approach.



R: RSS Fingerprint Database
Features
Labels
$RSS_1$
$RSS_2$
...... ......
$RSS_m$
B
F
X
Y
Feature Collector
Step 1: AP Confidence calculation
Step 2: P-T-SNE
Two-Step Reliable Feature Selector
Offline Stage
Building-aware attention
Building and Floor concat
Multi-Scale Feature Fusion Module
Multi-Task Positioning Model
Task 1: Building classification
Task 2: Floor classification
Task 3: Coordinate regression
Multi-Tasks Learning Module
Σ
Multi-Task Joint Learning
Online Stage
Network attacks
Environmental factors
Real-time RSS
Reliable Feature Selector
Multi-Task Positioning Model
Location(B,F,X,Y)

*Fig.1 Structure of Current Approach from [1]*

However, to accommodate our project's aim, we decided to amend this flow: Since we do not need to estimate for the building, we dropped the building classification model and used SVM to do floor classification. And there is no joint learning between floor and coordinate due to time constraints. Which means when the floor number has been classified out, it is used to wake the corresponding regression model based on its floor number and present the estimated coordinate. For the part of 'two step reliable feature selector', it is implemented with a slightly different version in and will be covered later part of this report.

This current method shows a good result in distance error of estimate coordinate, with an average of 1.7 meters of error. It is worth exploring and getting useful ideas from it.

## Related Concepts

### PCA (Principal Component Analysis)

PCA is a dimension reduction tool and feature selector widely adopted in machine learning method. Its idea is to perform eigendecomposition on the data's the covariance matrix to get the eigenvectors and eigenvalues and select the best principal component to perform dimensionality reduction. It has several advantages which could possibly benefit our task:

- Remove noise: WIFI signal is unstable and unreliable, it might be helpful if we could remove those interfering AP by selecting the 'k' best features.
- Reduce Training Cost: PCA will remove unimportant features and increase the dependence between the data, this could help the training converge faster.

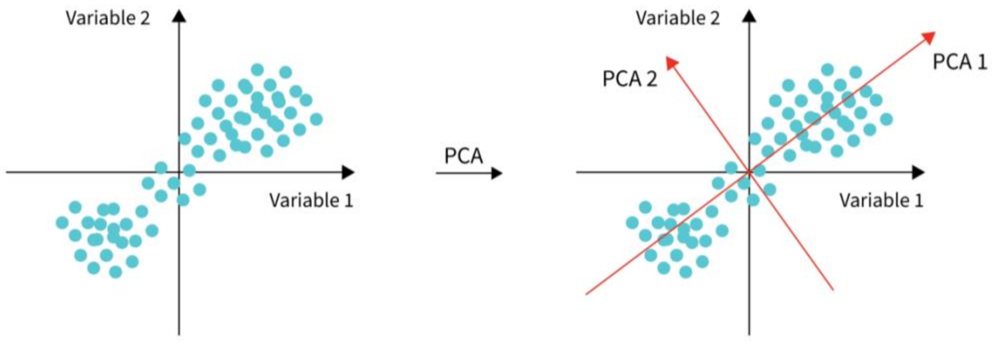


*Fig.2 PCA Conceptual Graph [3]*

### SNE (Stochastic Neighbour Embedding)

It is a non-linear dimension reduction method which can convert high dimensional dataset into 2-D or 3-D for visualization [4]. This method applies affine transformation from data points to probability distribution in two steps:

1. Construct probability distribution in original high dimensional space. Data with higher similarity will be selected.

2. Rebuild the probability distribution in lower dimensional space according to the beforehand distribution.

It converts Euclidean distance into conditional probability $p_{i|j}$ to present the similarity between data. $x_j$ and $x_i$ means two data points coordinate in the same dataset in given space.

$$\mathbf{p_{j|i}} = \frac{\exp\left(-\|x_i - x_j\|^2/(2\sigma_i^2)\right)}{\sum_{k\neq i}\exp\left(-\|x_i - x_k\|^2/(2\sigma_i^2)\right)} \quad (1)$$

$$q_{j|i} = \frac{\exp\left(-\|x_i - x_j\|^2\right)}{\sum_{k\neq i}\exp\left(-\|x_i - x_k\|^2\right)} \quad (2)$$

For lower dimensional space, we define the variance under Gaussian Distribution to be 1/sqrt(2). The similarity is presented in $q_{j|i}$.

We want to optimize the *Kullback-Leibler divergences* between two distributions to make $p_{i|j} = q_{j|i}$. Construct a cost function to perform optimization learning.

$$\boldsymbol{Cost} = \sum_i \boldsymbol{KL}(\boldsymbol{P_i} \parallel \boldsymbol{Q_i}) = \sum_i \sum_j \boldsymbol{p_{j|i}} \log \frac{p_{j|i}}{q_{j|i}} \quad (3)$$

It is important to notice that due the asymmetric property of KL-divergences, the optimization will punish heavier when the farther data pairs in new space represent the closer data pairs in original space, but less vice versa. Hence, SNE will tend to contain the local feature rather than global feature of the dataset.

For the $\sigma_i$ in equation (1), a concept called perplexity is introduced as a hyper-parameter to search for the best $\sigma_i$. While 'H' refers to entropy. Then, compute the gradient of cost function, y here means the mapping point in new space.

$$\mathrm{Perp}(P_i) = 2^{H(P_i)} \qquad \frac{\delta C}{\delta y_i} = 2\sum_j \left(p_{j|i} - q_{j|i} + p_{i|j} - q_{i|j}\right)\left(y_i - y_j\right)$$

**t-SNE (t-Distributed Stochastic Neighbour Embedding)**

However, this approach exists a so called ‘crowding problem’ while points tend to assemble in low dimensional space. Later, Hinton and Maaten published a new method called t-SNE [5], with a symmetric SNE and using t-distribution to replace the pre-defined Gaussian distribution to tackle with the ‘crowding problem’.

T-distribution is remarkably similar to Gaussian distribution but with a much heavy-tailed under low degree of freedom. And we change the joint probabilities $q_{ij}$ as follows:

$$q_{ij} = \frac{\left(1 + \|y_i - y_j\|^2\right)^{-1}}{\sum_{k \neq l} \left(1 + \|y_k - y_l\|^2\right)^{-1}}$$

And resulting the gradient optimization equation become even simpler:

$$\frac{\delta C}{\delta y_i} = 4 \sum_j (p_{ij} - q_{ij})(y_i - y_j)\left(1 + \|y_i - y_j\|^2\right)^{-1}$$

In this task, we have adopted both dimension reduction algorithms, and compare with the training result of the unchanged dataset. To analyze whether dimensionality reduction or say feature extraction plays a key role in building an effective and correct prediction model with limited and flawed dataset.

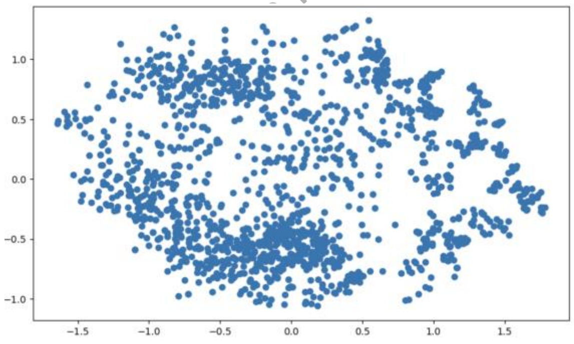


*Fig.3 L2 data distribution after applying PCA*

This graph no clear clusters are being formed. Data is spread widely, and each point remains some degree of independence.

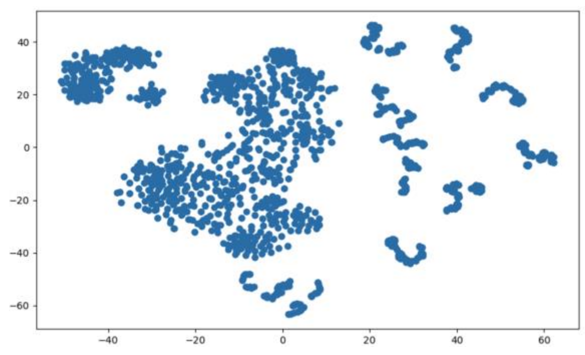


*Fig.4 L2 data distribution after applying t-SNE*

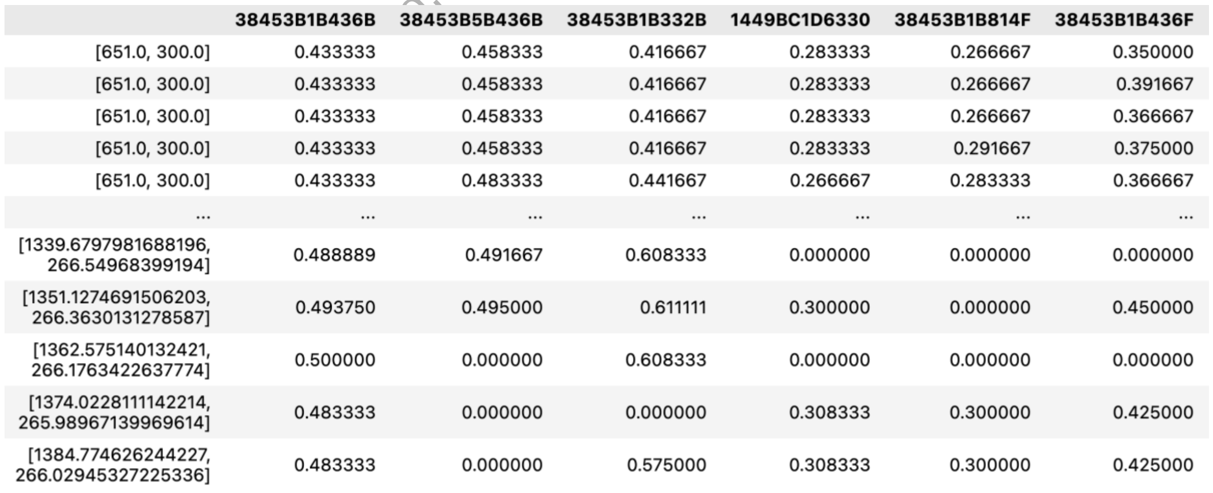
This graph shows some clusters have been formed after several t-SNE iterations.

## Experience Setup

### Data Collection

The desired input data format is {mac: RSS value}, we discard all irrelevant data such as frequency or the name of the WIFI. And the corresponding ground truth is in the format {x-coordinate, y-coordinate}. One location pair may consist of more than one mac address pair. The dataset given holds two sources, and one source consists of 5 floors' data (GF, L2, L4, L5, U1), while the second source consists of 3 floors' data (L2, L4, L5). Since we discover no significant difference in data quality and variation in the floor of GF and U1 compared with the rest. We decided to build a model for floor L2, L4 and L5, due to larger in data size, and hence outcomes with a better performance of the model.

We extract the data from the original text file, frame them into 3 matrixes, each row represents a coordinate pair, each column stands for a mac address, and for each intersection, it contains the RSS value corresponding to the row and column. Also, the max-min normalization (on scale 0-120, reverse the negative signal value) has been performed to the RSS value at the same time.



| | 38453B1B436B | 38453B5B436B | 38453B1B332B | 1449BC1D6330 | 38453B1B814F | 38453B1B436F |
|---|---|---|---|---|---|---|
| [651.0, 300.0] | 0.433333 | 0.458333 | 0.416667 | 0.283333 | 0.266667 | 0.350000 |
| [651.0, 300.0] | 0.433333 | 0.458333 | 0.416667 | 0.283333 | 0.266667 | 0.391667 |
| [651.0, 300.0] | 0.433333 | 0.458333 | 0.416667 | 0.283333 | 0.266667 | 0.366667 |
| [651.0, 300.0] | 0.433333 | 0.458333 | 0.416667 | 0.283333 | 0.291667 | 0.375000 |
| [651.0, 300.0] | 0.433333 | 0.483333 | 0.441667 | 0.266667 | 0.283333 | 0.366667 |
| ... | ... | ... | ... | ... | ... | ... |
| [1339.6797981688196, 266.54968399194] | 0.488889 | 0.491667 | 0.608333 | 0.000000 | 0.000000 | 0.000000 |
| [1351.1274691506203, 266.3630131278587] | 0.493750 | 0.495000 | 0.611111 | 0.300000 | 0.000000 | 0.450000 |
| [1362.575140132421, 266.1763422637774] | 0.500000 | 0.000000 | 0.608333 | 0.000000 | 0.000000 | 0.000000 |
| [1374.0228111142214, 265.98967139969614] | 0.483333 | 0.000000 | 0.000000 | 0.308333 | 0.300000 | 0.425000 |
| [1384.774626244227, 266.02945327225336] | 0.483333 | 0.000000 | 0.575000 | 0.308333 | 0.300000 | 0.425000 |

*Fig.5 formalized input data with RSS value normalization.*

**Confidence Calculation Model**

The mechanism of this model is to assign a confidence value on each of the coordinate pair, if the value is higher, which means the APs (accessing points) in this location are more trustworthy. This is done by:

$$C_j = \frac{1}{n}\sum_{i=1}^{n}\left(RSS_{j,i} - \frac{1}{n}\sum_{i=1}^{n} RSS_{j,i}\right)^{2}$$

Where $C_j$ is the confidence value of jth AP, and $RSS_{j,i}$ means the RSS value of $j^{th}$ AP at $i^{th}$ location. In another sense of this confidence value is the measurement of interclass dispersion of n-type samples. And after getting the values, we apply another min-max scaler to the values. And finally, it becomes the sample weight of the input training dataset. This step is important as it defines what kind of a list of data the neural network should pay more attention to. RSS value does not fluctuate a lot, so it is hard for the model to perform prediction based such dataset. The confidence value makes sense as it captures the variation of a single data, more variation means a sharper feature that could be learned.

**Coordinate Regression Model**

As verified, the number of neurons and layers does not to be large in this simple regression task. A dropout layer is added to prevent overfitting. The activation function choosing is arguable, we have tried using ‘relu’ or ‘leaky relu’, but the validation loss is lower when the activation function is set to be ‘tanh’. A likely reason is our regression is simple in structure and the gradient transmission between layers are still in a valid range. Plus, using ‘tanh’ might help to a more elaborated learning. The use of ‘log-cosh’ function is like the mean-square-error function, but with a better toleration to some large error caused by some ‘outliers’ [6]. This is suitable in our task since APs are not that trustable, WIFI signal undulation is common in real world applications.

## Implementation Details

| **Dimensionality reduction tool** | **n_components** | **perplexity** |
|---|---|---|
| PCA | 100 | n/a |
| t-SNE | 3 | 29 |

*Table.1 Showing different dimensionality reduction tools in use with parameters.*

| **Coordinate Regression Model** | **Values** |
|---|---|
| Linear Layer | 124 activation function: tanh |
| Dropout Layer | 10% |
| Linear Layer | 16 |
| Output Layer | 2 |
| Loss Function | log-cosh |

*Table.2 Showing the structure of the regression model and its inside parameters.*

| **Training Parameters** | **Values** |
|---|---|
| validation split | 0.3 |
| batch size | 4 |
| sample weight | confidence value obtained before |
| optimizer | Adam |
| learning rate | 0.005 |

*Table.3 Showing the parameters setting during training.*

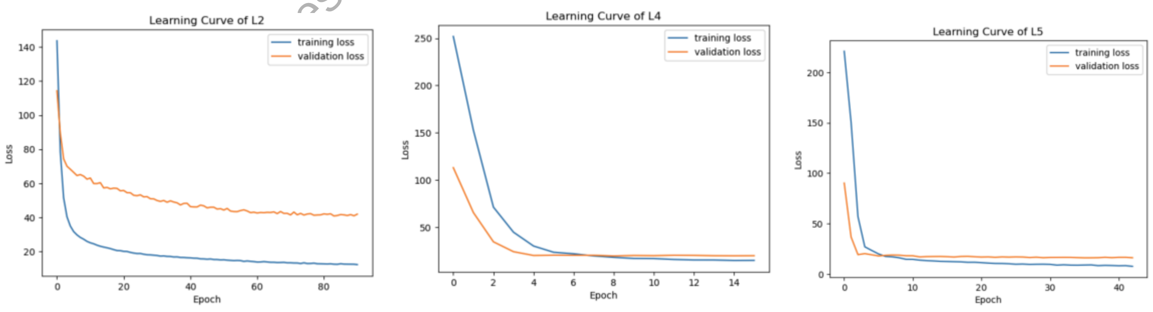


*Fig.6 The learning curve of three floors*

## Result Analysis

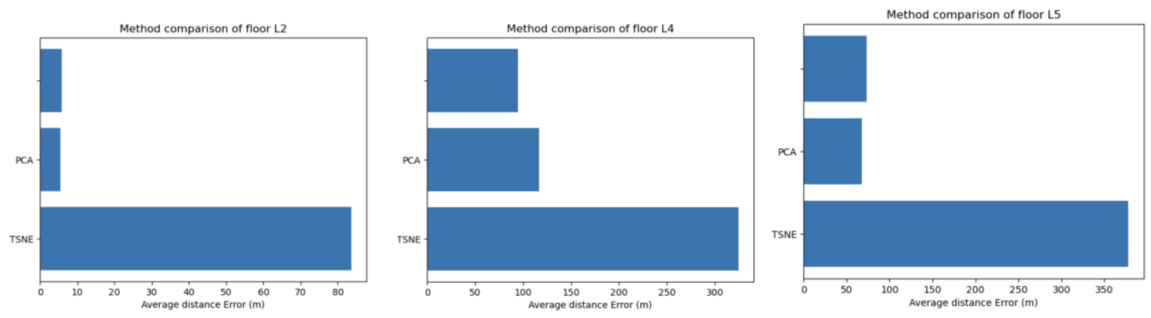


*Fig.7 Error of three floors' prediction with comparing different algorithm used:*

*(From left: no algorithm, PCA, t-SNE)*

We calculate the average distance error between predicted and true coordinate where the input is a small portion extracted from the original dataset (5%, non-trained).

The result showing is daunting expect the prediction for L2 using PCA or using no extra algorithm. The error could be controlled within 5 meters, which reaches the level of normal accuracy.

It is noticable that the error of L4 and L5 is way higher than L2. Possible reasons are because of the limit dataset size of floor L4 and L5, plus the floor area is larger in L4 and L5 (doubled compared to L2), which makes the model harder to output an extact location. More important factor is that there is more AP in floor L2, more AP means more learnable feature for the model during training. This result reflects that dataset's size and quality could affect the prediction performance heavily.

If we compare the two-dimension reduction tools, we find that PCA outperforms t-SNE to a large extent. And for floor L2 and L5, using PCA favors the model's performance. The question is, t-SNE is a powerful tool which usually supplies solid results to the performance, why it fails in our task? One possible guess is that, referring to Fig.4, the graph formed by t-SNE shows more clusters than PCA's graph. This phenomenon reflects the fact that t-SNE tends to maintain the local features when it rebuilds datapoints in lower dimension, but its cost is the loss of generality. However, regression models require more global features.

Features forming a cluster may be a good sign in training a classification model, but it is not favoring the regression model from our result.

Then we did a prediction with extra points on a 2-D map of floor L2. The green dots are the correct location, they are extracted from two paths where our signal collector walks from left to right in this map. While the blue dots are the predictions.

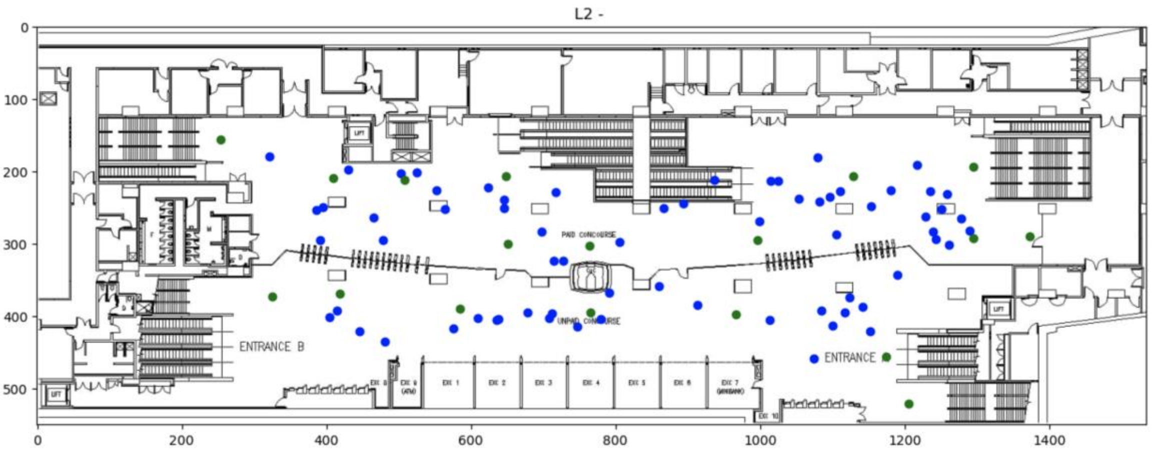


*Fig.8 Prediction on the map of L2 with current best approach*

This result is quite promising, as we can see those blue dots are forming two paths estimating the person is walking from left to right.

## Conclusion & Outlook

To build a powerful and accurate indoor localization estimating model requires many things: a large enough and high-quality dataset, suitable dimensionality reduction algorithm, correct sample weighting and a promising machine learning approach. For example, we can adopt grid-classification method to draw out which coordinate grid the user is standing on. Or we can fuse different input data, such as using WIFI together with Bluetooth fingerprints to perform a combined, multimodal learning. We believe that it is possible to make a better model if more time and hardworking are given.

## References

[1] Faheem Zafari, Athanasios Gkelias and Kin K. Leung “A Survey of Indoor Localization Systems and Technologies”, in IEEE COMMUNICATIONS SURVEYS & TUTORIALS, VOL. 21, NO. 3, THIRD QUARTER 2019

[2] Chun Wang, Juan Luo and Xiangjian He “Secure and Reliable Indoor Localization Based on Multitask Collaborative Learning for Large-Scale Buildings”, in IEEE INTERNET OF THINGS JOURNAL, VOL. 9, NO. 22, 15 NOVEMBER 2022

[3] Principal Component Analysis (PCA) , scaler.com
https://www.google.com/url?sa=i&url=https%3A%2F%2Fwww.scaler.com%2Ftopics%2Fnlp%2Fwhat-ispca%2F&psig=AOvVaw3sd8JpGiB0GkvGS8z6A3ap&ust=1690893772100000&source=images&cd=vfe&opi=89978449&ved=0CBEQjRxqFwoTCIi_kqf8uIADFQAAAAAdAAAAABAE

[4] Geoffrey Hinton and Sam Roweis, “Stochastic Neighbor Embedding”, 2002

[5] Laurens van der Maaten and Geoffrey Hinton, “Visualizing Data using t-SNE”, Journal of Machine Learning Research 9 (2008)

[6] Resve A. Saleh and A.K.Md. Ehsanes Saleh, “Statistical Properties of the log-cosh Loss Function Used in Machine Learning”, 2022