Machine Learning Based Distance Estimation Using RSSI Measurements

YIFAN YE WEAAM BAYAA

yifany|bayaa@kth.se

October 26, 2018

Abstract

With the development of the Internet of Things and proximity-based services paradigms, indoor proximity detection has become an interesting research topic. Received Signal Strength Indicator (RSSI) has frequently been used in distance estimation due to its availability without the need for investment in new infrastructure. The tradition way to estimate distance is based on the signal propagation model. However, fluctuations of RSSI in complicated environments reduce RSSI' s accuracy of distance estimation. In this research, we investigate distance estimation in an indoor environment with the use of machine learning (ML) methodologies. Three ML algorithms (Decision Tree Classifier, Random Forest Classifier, and K- Nearest Neighbor) are used to estimate the distance of devices from a WiFi access point (AP). Rather than using the RSSI values directly in the modelling, statistical properties (mean, standard deviation, minimum, maximum) of RSSI measurements have been used to enhance the accuracy of estimation. The accuracy of the proposed models are evaluated, experimental results show that the use of statistical properties can decrease negative effect of RSSI measurements' uncertainty, and compared with the propagation-based model, ML-based models improve the accuracy of 15.71%-18.87%. Among three ML models, random forest performs best.

keywords: Distance Estimation, Wi-Fi, RSSI, IoT, Proximity-based Services, Machine Learning.

Contents

List of Acronyms and Abbreviations

RSSI	received signal strength indicator
ML	machine learning
AP	access point
LNSM	Log-normal shadowing model
DTC	decision tree classifier
RFC	random frost classifier
KNN	K- Nearest Neighbor
RDS	raw dataset
ADE	average distance error

1 Introduction

Many services are proposed to leverage from IoT infrastructure. Location Based Services (LBS) and Proximity Based services (PBS) are examples of such services. Proximity detection of users and devices have wide-scale applications in many sectors. Smart homes/offices are applications where appliances or personal computers can be turned on/off according to user proximity. Interactive libraries/museums is another example where information is provided by playing sounds or videos according to the proximity of users.

Before discussing different technologies and techniques used for distance estimation, it is important to differentiate between positioning and proximity detection. Proximity detection is the process of estimating the distance between a user, and a Point of Interest [1] . For the outdoor environment, the Global Position System (GPS) has been widely used for positioning and distance estimation applications. Since the GPS relies mainly on signal propagation in the air, the accuracy of the GPS is degraded severely in indoor environments as the buildings' infrastructure will impact the propagation [2]. Accordingly, all efforts have been redirected to other technologies such as WiFi, Radio Frequency Identification (RFID), Ultra-wideband (UWB), ultrasound, Zigbee, BLE, etc. [3]. Many metrics can be used for this purpose such as RSSI, Angle of Arrival (AoA), Time of Flight (ToF), Time Difference of Arrival (TDoA) etc. [4]. Most of the existing devices are WiFi enabled which makes WiFi an attractive technology for many researchers. The traditional way of RSSI-based ranging is based on signal propagation models like the log-normal shadowing model (LNSM), which gives a relationship between RSSI and distance. LNSM is simple, cost-efficient, and quite accurate for free space [4, 5], however, it suffers from poor accuracy indoors due to additional signal attenuation resulting from the existence of obstacles, and severe RSSI uncertainty due to interference, and the distance estimation error can be up to 50% [5–7].

In this research, we propose a novel ML-based method to enhance the accuracy of indoor distance estimation. We exploit the statistical properties of RSSI measurements to decrease the influence of RSSI' s uncertainty. And we use three ML algorithms: Decision Tree Classifier (DTC), Random Forest Classifier (RFC), and K- Nearest Neighbor (KNN). The remainder of this research is constructed as follows: section 2 and 3 analyze the related work and research goals; section 4 presents the proposed method of using ML algorithms on different datasets; results and discussions are shown in section 5; finally, conclusions are listed in section 6.

2 Related work

Many kinds of research have been conducted to improve the accuracy of indoor distance estimation, which benefit IoT services like proximity detection. Some researchers proposed modified LNSMs, which take attenuation and reflection of obstacles into account [8–11], but the attenuation and reflection factors of obstacles need to be measured first, and these models cannot cope with random interference and noise. Some researchers built ranging models using BP neural network, based on the Kolmogorov theorem that a 3-layer BP neural network can fit any continuous function [12–14], but tons of data need to be collected to train the neural network. Apple's Bluetooth Low Energy (BLE) based iBeacon solution primarily intends to provide Proximity detection [15]. BLE-enabled devices have an application to listen to RSSI in beacons to estimate the proximity to iBeacon device, a fundamental constraint of iBeacon is that only the average RSSI value is reported to the user device; Biehl et al. [16] present LoCo that uses WiFi AP. RSSI values which obtained from the WiFi APs are used to train a classifier during offline phase. During the online phase, the collected RSSI values on the user device are then used to estimate the user's probable location; Huang et al. [17] present WalkieLokie that relies on acoustic signals measurement on the user device to calculate the relative position. WalkieLokie requires the user device to be outside the pockets so that it can receive inaudible acoustic signals from specific radio nodes (RNs) or speakers, due to the use of acoustic signals, the range of the system is limited to less than 8 m. Willis et al. [18] present a passive RFID information grid that can assist blind users in obtaining location and proximity related information. The user shoe is integrated with an RFID reader that can communicate with user device using Bluetooth. An RFID tag grid, programmed with spatial and ambience related information, is placed on the ground so that the reader in user shoes can read the position related information and convey it to the blind users. Zhang et al. propose LiTell [19] that uses fluorescent lights as the RNs and the user device camera as the receiver. LiTell uses characteristic frequency to differentiate among different RNs and then localize different users based on their proximity to a certain RN. LiTell requires fluorescent lights, which might not be present everywhere.

This research focuses on using ML approaches to estimate distance. To reduce the impact of the uncertainty of RSSI values, mean, minimum, maximum, and standard deviation of RSSI measurements are used as input features for the ML algorithms. Then, different ML algorithms are investigated to compare accuracy.

3 Research goals

Despite the fact that LNSM results in low accuracy, we want to investigate if it still can be used for limited distance applications. Furthermore, we exploit the statistical properties of RSSI measurements to see if adverse effects of fluctuations of RSSI measurements can be reduced. Finally, we test three different ML-based models (DTC, KNN, and RFC) and LNSM to conclude the best performing model.

4 Method

The framework of our research is shown in figure 1. We collect RSSI measurements in a complex indoor environment to build an LNSM and three ML-based models; statistical feature vectors are fed into ML-based models to decrease the impact of RSSI' s uncertainty. The performance of these models will be evaluated and compared.

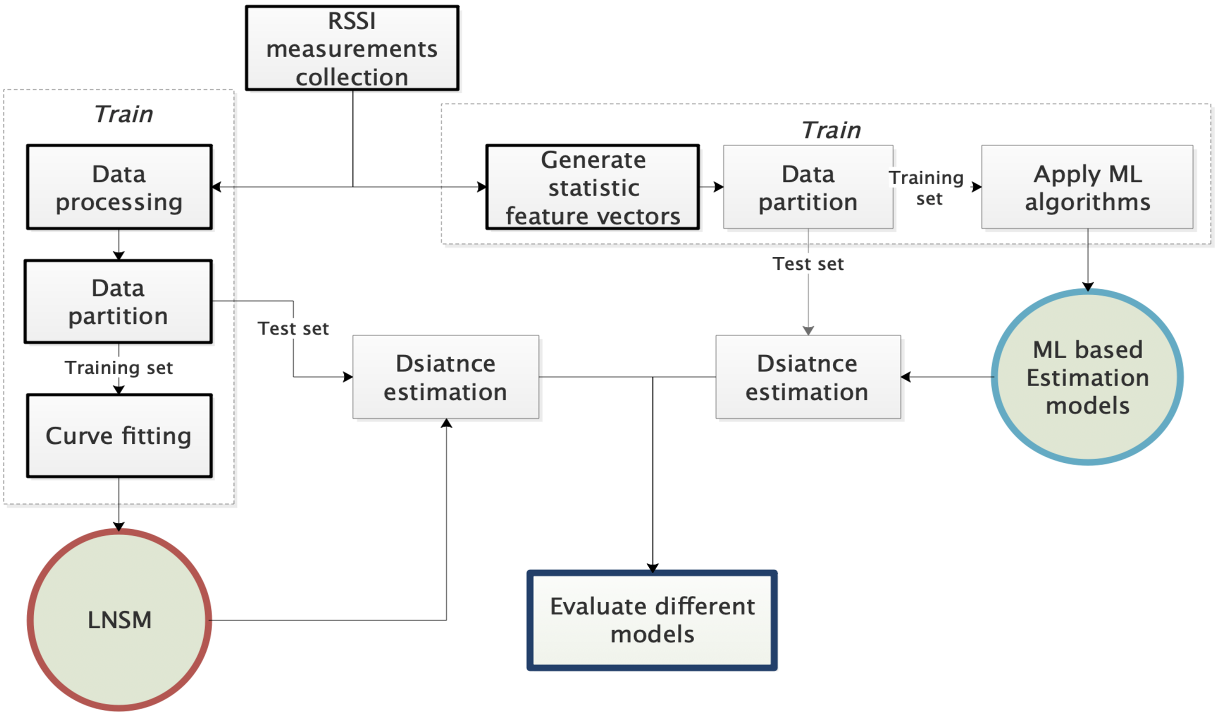
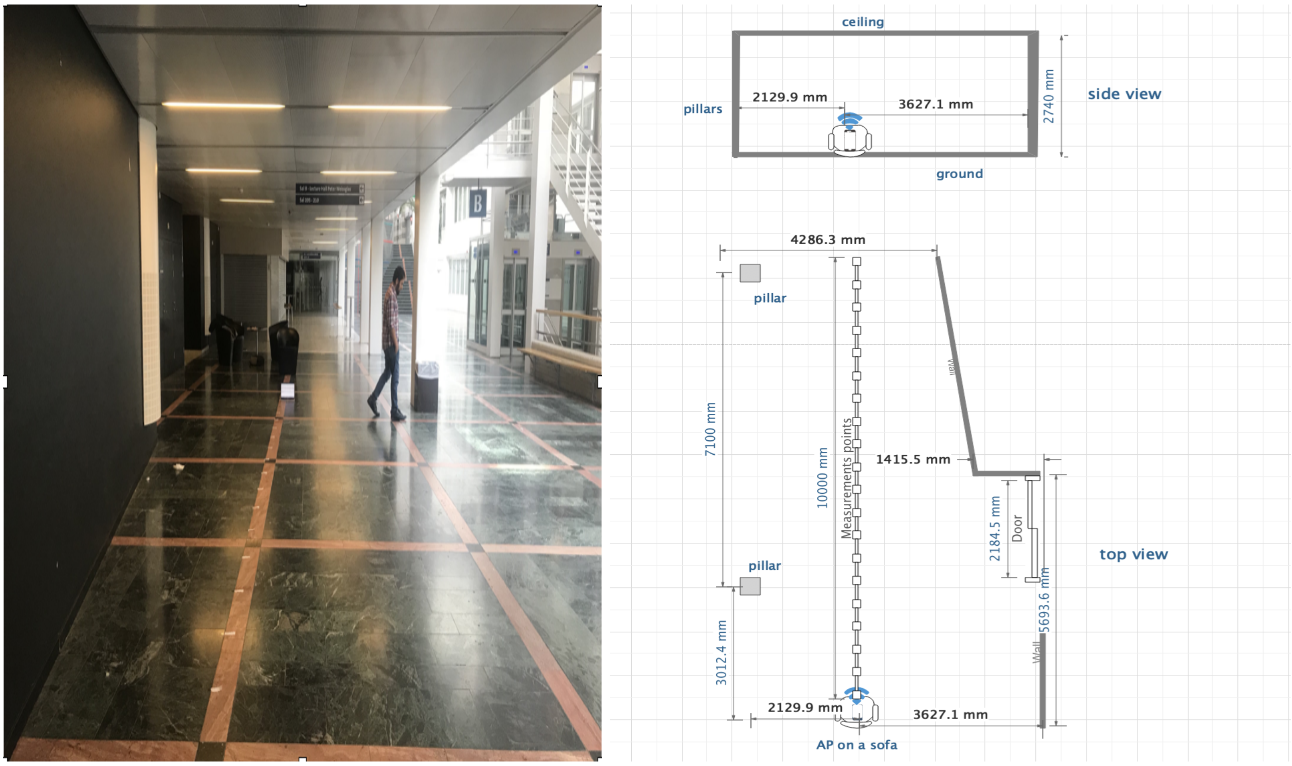


Figure 1: Framework.

4.1 Experiment setup

The experiment is conducted in the corridor of KTH campus in Kista with lots of sources of attenuation, reflection, absorption, and interference due to walls, obstacles, pillars, ceiling, and people that walked around. An Apple iPhone located on one sofa is used as an AP as shown in figure 2(a), and Wireshark is set as the monitoring mode to collect RSSI measurements.



(a) Scene.

(b) Floor plan.

Figure 2: Experimental situation.

4.2 Experiment data

As shown in figure 2(b), we have 21 sampling points, each point is defined in one dimension (horizontally), and the distance between each point is 0.5 meter. At each point, 3000 RSSI measurements are collected. The raw dataset (RDS) is shown in the equation (1).

$$RDS = \begin{bmatrix} (d_0, y_1^0) & (d_0, y_2^0) & (d_0, y_3^0) & \cdots & (d_0, y_n^0) \\ (d_1, y_1^1) & (d_1, y_2^1) & (d_1, y_3^1) & \cdots & (d_1, y_n^1) \\ (d_2, y_1^2) & (d_2, y_2^2) & (d_2, y_3^2) & \cdots & (d_2, y_n^2) \\ \vdots & \vdots & \vdots & \ddots & \vdots \\ (d_m, y_1^m) & (d_m, y_2^m) & (d_m, y_3^m) & \cdots & (d_m, y_n^m) \end{bmatrix} \tag{1}$$

Where m is the sequence number of sampling points $(m = 20)$, n refers to times that RSSIs are measured at each point$(n = 3000)$, d_i represents i-th point's distance from AP $(0 \leqslant i \leqslant m)$ is the j-th of measurements at point i $(0 \leqslant i \leqslant m, 1 \leqslant j \leqslant n)$.

4.3 Statistic feature vectors expression

RSSI measurements at the same point fluctuate drastically in a complex indoor environment full of noise and interference, but they tend to satisfy a certain distribution law [7,20]. Figure 3 gives an intuitive feeling of the uncertainty of RSSI measurements at sampling point 5. Feature vectors are created, and statistical property is calculated for each vector as illustrated in table 1, the table shows better stability of features.

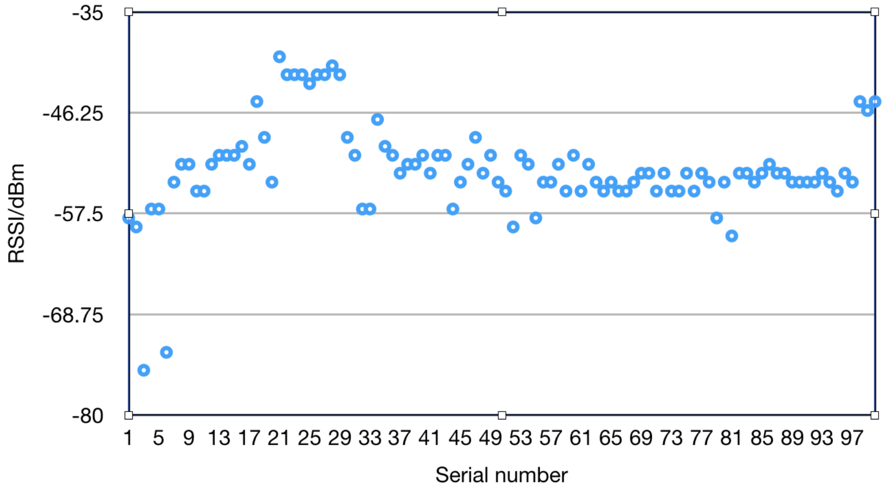


Figure 3: RSSI at point 5 change over time.

Table 1: Four Vectors at point 5.

Vector sequence number	Mean/dBm	Std/dBm	Maximum/dBm	Minimum/dBm
1	-52.51	5.2637	-40	-75
2	-53.30	3.2860	-44	-62
3	-54.01	3.8833	-48	-76
4	-54.45	3.4153	-48	-75

The statistical feature vectors are constructed out of RDS derived from equation (1), which are fed to ML algorithms. For every 100 consecutive RSSI measurements, the mean, standard deviation, maximum, and minimum are calculated to be assembled into a vector as is shown in equation (2).

$$\vec{V}_j^i = \begin{bmatrix} mean_j^i & std_j^i & max_j^i & min_j^i \end{bmatrix}, (0 \leqslant i \leqslant m, 1 \leqslant j \leqslant \frac{n}{100}) \tag{2}$$

Where :

$\vec{V}_j^i$— j-th feature vectors at i-th points,
$mean_j^i$— mean of j-th 100 consecutive RSSI measurements at i-th points,
std_j^i— standard deviation of j-th 100 consecutive RSSI measurements at i-th points,
max_j^i— maximum of j-th 100 consecutive RSSI measurements at i-th points,
min_j^i— minimum of j-th 100 consecutive RSSI measurements at i-th points.

4.4 LNSM-based approach

LNSM gives the relationship between RSSI and distance [21, 22] as is shown in equation (3)

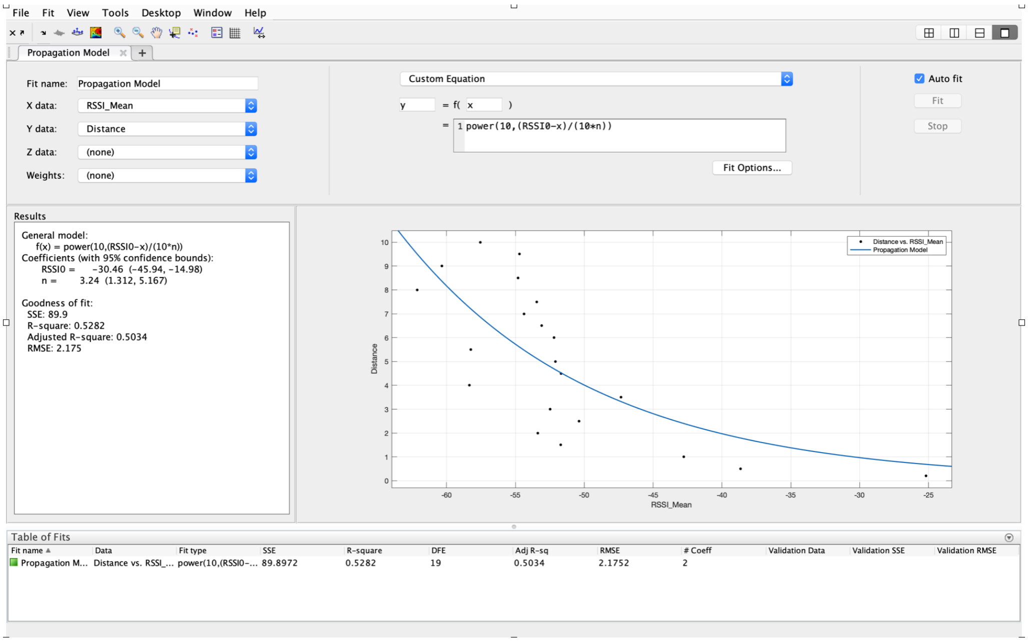
$$RSSI = -10nlg(d) + A \tag{3}$$

Then we have:

$$d = 10^{\frac{A-RSSI}{10n}} \tag{4}$$

where:
RSSI– measured value of the receiver, in [dBm],
d– distance between the receiver and AP , in [m],
n– propagation constant,
A – measured value of *RSSI* at reference distance 1m , in [dBm].

Using equation (4) to fit collected data, we can obtain the value of n and A , thereby a correlation between distance and RSSI is built. Matlab curve fitting tool [23] provides an easy way for curve fitting, as is shown in figure 4. We choose the mean of RSSI measurements at each point as X data, and distance as Y data, then customize the fitting equation the same as equation (4). The Matlab will fit the curve and give the parameter n and A automatically.



Fit name: Propagation Model
X data: RSSI_Mean
Y data: Distance
Z data: (none)
Weights: (none)

Custom Equation

y = f(x)
= power(10,(RSSI0-x)/(10*n))

```
General model:
    f(x) = power(10,(RSSI0-x)/(10*n))
Coefficients (with 95% confidence bounds):
      RSSI0 =     -30.46  (-45.94, -14.98)
      n =        3.24  (1.312, 5.167)

Goodness of fit:
  SSE: 89.9
  R-square: 0.5282
  Adjusted R-square: 0.5034
  RMSE: 2.175
```

Fit name ▲	Data	Fit type	SSE	R-square	DFE	Adj R-sq	RMSE	# Coeff	Validation Data	Validation SSE	Validation RMSE
🟩 Propagation M...	Distance vs. RSSI_...	power(10,(RSSI0-...	89.8972	0.5282	19	0.5034	2.1752	2			

Figure 4: Matlab curve fitting toolbox.

4.5 ML-based approach

The ML approach consists of two stages: the first stage is the training stage where data is collected and used as input to a classifier to build a model, and the second stage is called the testing phase, where new data are tested against the model. Three classical ML classification algorithms are adopted in our research: DTC, RFC, and KNN.

DTC is a popular machine learning classification algorithm with low algorithm complexity. A decision tree consists of internal nodes, leaf, and directed edges. Every internal node represents a test condition that is used to separate data, and one leaf represents one category. After we successfully build a DTC, it is easy to category data by letting data pass through the root all the way to a leaf [24].

RFC is an ensemble algorithm that evolves from DTC. In RFC, a collection of decision trees are generated, and the tree with the highest vote is chosen, RFC takes weights based on the input as a parameter that resembles the number of decision trees [25]. Those weights will be formed in the collaborative forest classifier without the conventional tree pruning process [26]. Many researchers have done some work to compare the performance between RFC and DTC, in most cases, RFC is less likely to overfit, thereby RFC performs better than DTC [26–29] , we also compare their performance on our dataset.

KNN is one of the easiest ML classifier algorithms that even "do not have train phase", the core of KNN is sample A belongs to the category which has the most samples among A' s k nearest neighbors [30]. The parameter k need to be set manually, in our research, we analyze the parameter k' s impact on the accuracy of distance estimation.

We use python sklearn library [31] to build DTC, RFC, and KNN models. The data is partitioned into a training set and test set firstly. There is no mature theory for deciding the exact ratio of splitting data, we adopted the common train/test ratio: 30/70 [25,32]. So the training set is an array with $420((2100/100)*21)$ 4-statistic-feature vectors, and a $420*1$ array which contains 21 different distance. The test dataset is an array with $199((900/100)*21)$ 4-statistic-feature vectors, and a $199*1$ array which contains 21 different distance.

4.6 Evaluation index

For the evaluation of the accuracy of different methods of distance estimation, we adopt average distance error (ADE) as is indicated in equation (5).

$$ADE = \sum_{i=1}^{n_t} \left(\left| \hat{d_i} - d_i \right| / n_t \right) \tag{5}$$

where $\hat{d_i}$ and d_i are estimation distance and real distance for i-th test data, and n_t refers to the size of test dataset. The smaller ADE is, the better accuracy the model has.

5 Results and Analysis

5.1 Parameter K' s impact on accuracy of KNN-based model

K is the only hyper-parameter of KNN algorithm, and it denotes how many nearest neighbors a sample will choose. Value of K can have a high impact on KNN algorithm, a way too small K can easily lead to overfitting, while the model may under-fit the dataset if the K is too large. We want to know what is the best K for our dataset. Based on equation (5), for K ranging from 2 to 29, we separately compute average distance error within 10m. We can see from figure 5 That KNN based model have the highest performance when $K = 3$.

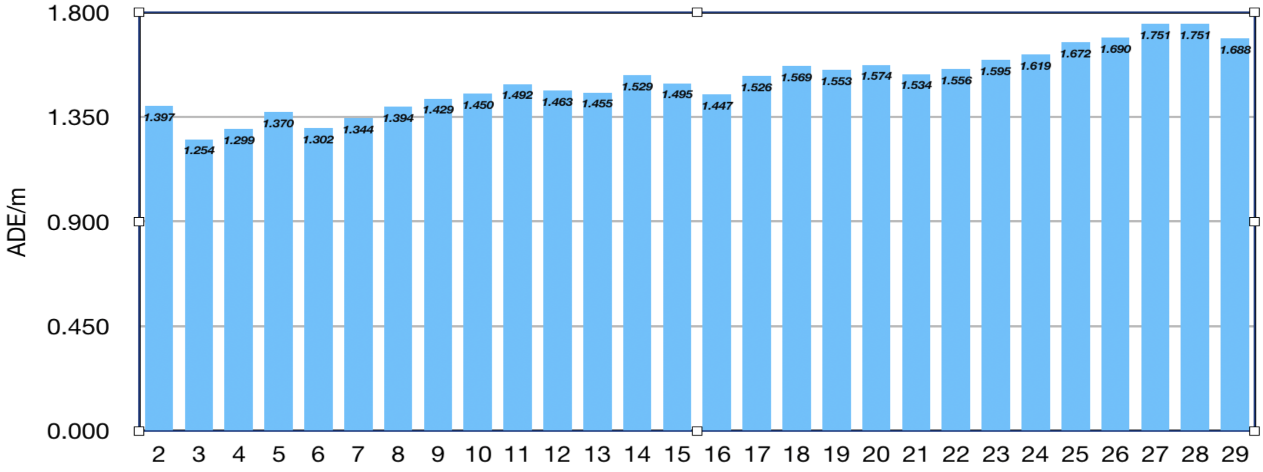


Figure 5: ADE - K.

5.2 Evaluation of statistical vectors' advantage and accuracy of the models

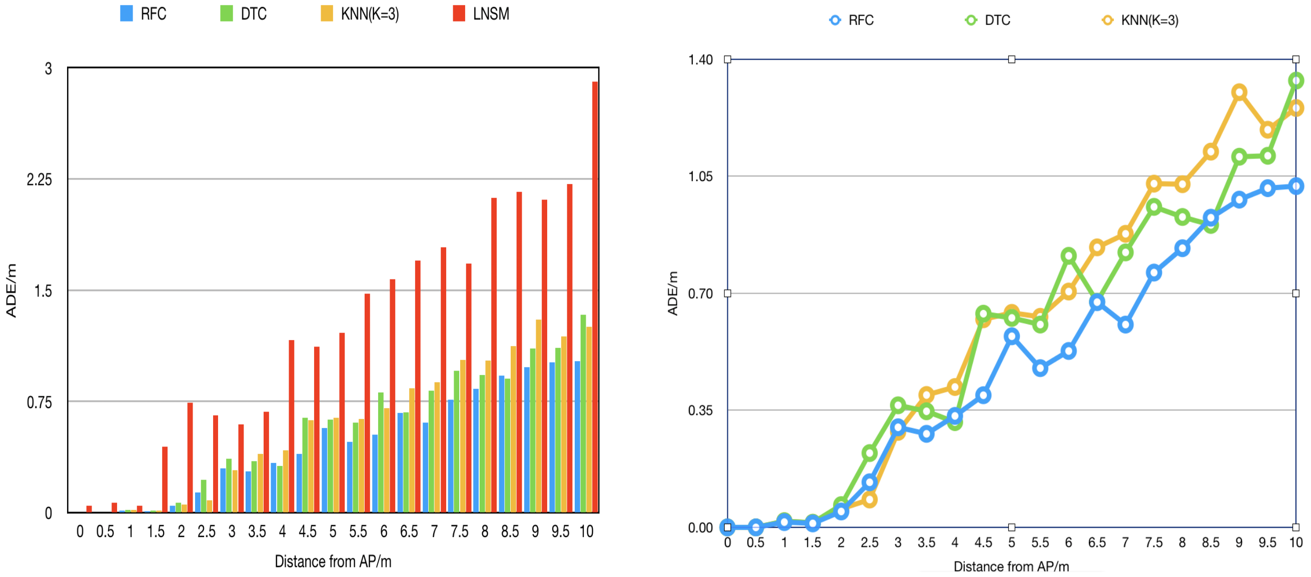
The train set is fed into 3 ML algorithms (KNN, DTC, RFC) and LNSM. Using equation (5), we calculate their ADE within 10m as in shown in table 2. For three ML-based models, two types of input data type (single RSSI, statistical vectors) are used to feed the ML algorithms, the results show that all three ML-based models have lower ADE with statistical vectors as input, which means the use of statistical vectors can reduce the influence of RSSI measurements' uncertainty, thereby improving the performance. RFC, DTC, and KNN improve the accuracy of 18.87%, 15.71% and 16.53% separately, compared with LNSM.

We also evaluate their accuracy within each distance as is shown in figure 6(a). Within any distance range, all the 3 ML-based models have less average distance error than the LNSM, and on the whole, accuracy decrease with the increase of distance. Near the AP, the accuracy of ML-based models is extremely high, while the propagation-based model fail to perform ideally, we think one reason is LNSM itself fails to work well in our experimental environment with much interference, the other reason is the space between every two sampling points is too long.

Figure 6(b) gives a clearer comparison of 3 ML-based models. The error curve of KNN and DTC almost overlap, and both of them have less accuracy than RFC, we think it is because that RFC overcomes overfitting of DTC to some degree [33], and " KNN often fails to perform well with an inappropriate choice of distance metric or due to numerous irrelevant features [34, 35] ".

Table 2: ADE comparision within 10m.

Input type	ADE/m			
	RFC	DTC	KNN	LNSM
Single RSSI	1.893	1.891	2.494	2.907
Statistical vectors	1.020	1.336	1.254	



(a) Comparison of 4 models.

(b) Comparison of 3 ML-based models.

Figure 6: ADE - Dsitance.

Some services like proximity detection can benefit from the above results. For devices that need high accuracy of distance estimation, RFC could be a better choice than LNSM. But with the improvement of accuracy, the consumption of computing complexity and memory increase at the same time, so the propagation-based model like LNSM is still an optional choice for devices that have limited memory and computational ability, and can tolerate a certain degree of estimation error.

6 Conclusion and future work

In this report, we have proposed a novel ML-based distance estimation scheme incorporating the use of the statistical feature vectors with mean, standard deviation, minimum and maximum values of RSSI measurements, and results show that it can decrease the uncertainty issue of RSSI measurements. Statistical feature vectors are used as an input to three ML algorithms (KNN, DTC, and RFC) to build three ML-based models, as a comparison, an LNSM is also built. Experimental results show that compared with the LNSM, our ML-based models can improve the accuracy of 15.71%-18.87% within 10m in our experimental scenario, and RFC performs best among three ML-based models.

In the future, we will explore more devices to collect and evaluate models, as different devices may receive different signal strengths from an AP at the same point, a model built from the dataset from one device may fail to accurately make a prediction for other devices. We also intend to look into deep neural networks and evaluate the level of accuracy, because we think that if much more data is collected in various environments, and more features are taken into account, the deep neural network could provide a better and more universal model. Furthermore, computational cost and needed memory can be evaluated for different ML algorithms.

7 Acknowledgements

This report is the final assignment of Research Methodology and Scientific Writing (II2202) at School of Information and Communication Technology, KTH Royal Institute of Technology. The research is conducted by Yifan Ye and Weaam Bayaa, we highly thank professor Gerald Q. Maguire Jr., Mr. Óttar Gudmun, and Mr. Andreas Dahm for their valuable suggestions and ideas, from which our research benefits greatly.

References

[1] F. Zafari, I. Papapanagiotou, M. Devetsikiotis, and T. J. Hacker, "Enhancing the accuracy of ibeacons for indoor proximity-based services," in *2017 IEEE International Conference on Communications (ICC)*, May 2017. doi: 10.1109/ICC.2017.7996508. ISSN 1938-1883 pp. 1–7.

[2] F. Subhan, H. Hasbullah, A. Rozyyev, and S. T. Bakhsh, "Indoor positioning in bluetooth networks using fingerprinting and lateration approach," in *2011 International Conference on Information Science and Applications*, April 2011. doi: 10.1109/ICISA.2011.5772436. ISSN 2162-9048 pp. 1–9.

[3] F. Zafari, I. Papapanagiotou, and K. Christidis, "Microlocation for internet-of-things-equipped smart buildings," *IEEE Internet of Things Journal*, vol. 3, no. 1, pp. 96–112, Feb 2016. doi: 10.1109/JIOT.2015.2442956

[4] B. Yang, Y. Lu, J. Wang, Y. Zhang, and Y. Ma, "An improved rbf neural network algorithm to mitigate the distance error based on rssi," in *2017 36th Chinese Control Conference (CCC)*, July 2017. doi: 10.23919/ChiCC.2017.8027945. ISSN 1934-1768 pp. 3759–3764.

[5] D. Gualda, J. Ureña, J. C. García, E. García, and D. Ruiz, "Rssi distance estimation based on genetic programming," in *International Conference on Indoor Positioning and Indoor Navigation*, Oct 2013. doi: 10.1109/IPIN.2013.6817881 pp. 1–8.

[6] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI," *ACM Computing Surveys*, vol. 46, no. 2, pp. 1–32, nov 2013. doi: 10.1145/2543581.2543592. [Online]. Available: https://doi.org/10.1145%2F2543581.2543592

[7] Q. Luo, X. Yan, J. Li, Y. Peng, Y. Tang, J. Wang, and D. Wang, "Dedf: lightweight wsn distance estimation using rssi data distribution-based fingerprinting," *Neural Computing and Applications*, vol. 27, no. 6, p. 1567–1575, 2015. doi: 10.1007/s00521-015-1956-2

[8] E. Damosso, L. Correia, I. M. European Commission. DGX III "Telecommunications, and E. of Research.", *COST Action 231: Digital Mobile Radio Towards Future Generation Systems : Final Report*, ser. EUR (Series). European Commission, 1999. [Online]. Available: https://books.google.se/books?id=setUHQAACAAJ

[9] J. H. Tarng and T. R. Liu, "Effective models in evaluating radio coverage on single floors of multifloor buildings," *IEEE Transactions on Vehicular Technology*, vol. 48, no. 3, pp. 782–789, May 1999. doi: 10.1109/25.764994

[10] S. Y. Seidel and T. S. Rappaport, "914 mhz path loss prediction models for indoor wireless communications in multifloored buildings," *IEEE Transactions on Antennas and Propagation*, vol. 40, no. 2, pp. 207–217, Feb 1992. doi: 10.1109/8.127405

[11] A. J. Motley and J. M. P. Keenan, "Personal communication radio coverage in buildings at 900 mhz and 1700 mhz," *Electronics Letters*, vol. 24, no. 12, pp. 763–764, June 1988. doi: 10.1049/el:19880515

[12] X. Chen, M. Zhang, K. Ruan, C. Gong, Y. Zhang, and S. X. Yang, "A ranging model based on BP neural network," *Intelligent Automation Soft Computing*, vol. 22, no. 2, pp. 325–329, nov 2015. doi: 10.1080/10798587.2015.1095484. [Online]. Available: https://doi.org/10.1080/10798587.2015.1095484

[13] Z. Xuhui, J. Junfeng, W. Jiaxin, and Z. Yingjie, "Using bp neural network algorithm improve the prediction accuracy of distance between zigbee," in *2011 Fourth International Conference on Intelligent Computation Technology and Automation*, vol. 2, March 2011. doi: 10.1109/ICICTA.2011.485 pp. 790–794.

[14] H. Zhang and X. Shi, "A new indoor location technology using back propagation neural network to fit the rssi-d curve," in *Proceedings of the 10th World Congress on Intelligent Control and Automation*, July 2012. doi: 10.1109/WCICA.2012.6357843 pp. 80–83.

[15] A. Inc, "ibeacon." [Online]. Available: https://developer.apple.com/ibeacon/

[16] J. T. Biehl, M. Cooper, G. Filby, and S. Kratz, "Loco: A ready-to-deploy framework for efficient room localization using wi-fi," in *Proceedings of the 2014 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '14. New York, NY, USA: ACM, 2014. doi: 10.1145/2632048.2636083. ISBN 978-1-4503-2968-2 pp. 183–187. [Online]. Available: http://doi.acm.org/10.1145/2632048.2636083

[17] W. Huang, X.-Y. Li, Y. Xiong, P. Yang, Y. Hu, X. Mao, F. Miao, B. Zhao, and J. Zhao, "Walkielokie: Sensing relative positions of surrounding presenters by acoustic signals," in *Proceedings of the 2016 ACM International Joint Conference on Pervasive and Ubiquitous Computing*, ser. UbiComp '16. New York, NY, USA: ACM, 2016. doi: 10.1145/2971648.2971655. ISBN 978-1-4503-4461-6 pp. 439–450. [Online]. Available: http://doi.acm.org/10.1145/2971648.2971655

[18] S. Willis and S. Helal, "A passive rfid information grid for location and proximity sensing for the blind user," *University of Florida Technical Report*, pp. 1–20, 2004.

[19] C. Zhang and X. Zhang, "Litell: Indoor localization using unmodified light fixtures: Demo," in *Proceedings of the 22Nd Annual International Conference on Mobile Computing and Networking*, ser. MobiCom '16. New York, NY, USA: ACM, 2016. doi: 10.1145/2973750.2985612. ISBN 978-1-4503-4226-1 pp. 481–482. [Online]. Available: http://doi.acm.org/10.1145/2973750.2985612

[20] Q. Luo, Y. Peng, X. Peng, and A. Saddik, "Uncertain data clustering-based distance estimation in wireless sensor networks," *Sensors*, vol. 14, no. 4, pp. 6584–6605, apr 2014. doi: 10.3390/s140406584. [Online]. Available: https://doi.org/10.3390%2Fs140406584

[21] R. K. Mahapatra and N. S. V. Shet, "Experimental analysis of rssi-based distance estimation for wireless sensor networks," in *2016 IEEE Distributed Computing, VLSI, Electrical Circuits and Robotics (DISCOVER)*, Aug 2016. doi: 10.1109/DISCOVER.2016.7806221 pp. 211–215.

[22] Ł. Chruszczyk and A. Zajac, "Comparison of indoor/outdoor, RSSI-based positioning using 433, 868 or 2400 MHz ISM bands," *International Journal of Electronics and Telecommunications*, vol. 62, no. 4, pp. 395–399, dec 2016. doi: 10.1515/eletel-2016-0054. [Online]. Available: https://doi.org/10.1515%2Feletel-2016-0054

[23] "Curve fitting toolbox." [Online]. Available: https://www.mathworks.com/products/curvefitting.html

[24] J. Elder, "Top 10 data mining mistakes," *Fifth IEEE International Conference on Data Mining (ICDM05)*, 2005. doi: 10.1109/icdm.2005.83

[25] S. Suthaharan, "Machine learning models and algorithms for big data classification: thinking with examples for effective learning." Springer, 2016. ISSN 978-1-4899-7641-3

[26] L. Breiman, "Random forests," *Machine Learning*, vol. 45, no. 1, pp. 5–32, Oct 2001. doi: 10.1023/A:1010933404324. [Online]. Available: https://doi.org/10.1023/A:1010933404324

[27] J. Ali, R. Khan, N. Ahmad, and I. Maqsood, "Random forests and decision trees," *International Journal of Computer Science Issues (IJCSI)*, vol. 9, no. 5, p. 272, 2012.

[28] M. Pal, "Random forest classifier for remote sensing classification," *International Journal of Remote Sensing*, vol. 26, no. 1, p. 217–222, 2005. doi: 10.1080/01431160412331269698

[29] A. Liaw, M. Wiener *et al.*, "Classification and regression by randomforest," *R news*, vol. 2, no. 3, pp. 18–22, 2002.

[30] M.-L. Zhang and Z.-H. Zhou, "Ml-knn: A lazy learning approach to multi-label learning," *Pattern Recognition*, vol. 40, no. 7, p. 2038–2048, 2007. doi: 10.1016/j.patcog.2006.12.019

[31] "Documentation of scikit-learn 0.20.0¶." [Online]. Available: http://scikit-learn.org/stable/documentation.html

[32] N. Abhishek, "Machine learning: A gentle introduction. – towards data science," Sep 2018. [Online]. Available: https://towardsdatascience.com/machine-learning-a-gentle-introduction-17e96d8143fc

[33] "Decision trees – handling overfitting using forests," Jun 2015. [Online]. Available: https://www.edupristine.com/blog/handling-overfitting-using-forests-in-decision-trees

[34] X. Liang, X. Gou, and Y. Liu, "Fingerprint-based location positoning using improved knn," in *2012 3rd IEEE International Conference on Network Infrastructure and Digital Content*, Sept 2012. doi: 10.1109/ICNIDC.2012.6418711. ISSN 2374-0272 pp. 57–61.

[35] R. Min, D. A. Stanley, Z. Yuan, A. Bonner, and Z. Zhang, "A deep non-linear feature mapping for large-margin knn classification," in *2009 Ninth IEEE International Conference on Data Mining*, Dec 2009. doi: 10.1109/ICDM.2009.27. ISSN 1550-4786 pp. 357–366.

A Appendix

The codes and original data set are given by the following hyperlink:
`https://github.com/yyfhust/RSSI-distance-estimation.git`