A Graph-Based Model for Leveraging Spatial and Fingerprint Similarity in RF-based Indoor Positioning

Liam Mark Self[1,*], Ian James Wassell[1]

[1]Department of Computer Science and Technology, University of Cambridge, United Kingdom

Abstract

A limitation on conventional deep learning methods for fingerprint-based indoor positioning is the difficulty in leveraging the underlying geometry of the environment. Solutions to this involving Graph Neural Networks have been proposed, however these place a dominant focus on the geometric relationships between reference points (RPs), which are often sparse in number, and ignore the relationships between the training data.

In this work, we present a proof-of-concept model for RF-based indoor positioning based on Graph Neural Networks that captures both the spatial relationships between RPs and the similarity between fingerprints of the training database. We use a Graph Attention Network to train a graph composed of 31 000 power readings gathered from a large office simulation of a environment produced using a commercial ray-tracer, and a Graph Convolutional Network to model the spatial relationships between RPs.

Preliminary evaluation shows a positioning accuracy of 0.78 m, a 29% improvement over benchmark models using kNN, MLP and Random Forest, and suggests that combining both graph models outperforms the use of either model independently.

Keywords

Indoor positioning, Fingerprinting, Graph Neural Networks, Data augmentation

1. Introduction and background

The rise of applications that require reliable, robust location information has increased rapidly in recent times, and location and context-aware technology transcends numerous fields and areas, from autonomous robotics[1], inventory management[2], interactive multimedia [3], through to the monitoring of the health of vulnerable adults[4, 5].

While GNSS technologies such as GPS are considering the 'gold standard' of modern positioning systems, their reliance on line-of-sight communication with satellites limits their practicality in indoor environments. RF-based indoor positioning systems, then, often turn to terrestrial technologies to provide reference points with which to locate devices, such as Bluetooth, Wi-Fi and Ultra Wide-Band (UWB). The use of Wi-Fi, for example, is nowadays ubiquitous for network and Internet communications, and while, for example, utilising channel state information (CSI) of wideband technologies like UWB can result in very high precision position estimation (in ideal conditions), Wi-Fi base stations remain a popular reference source due to the ability to leverage existing infrastructure. Information such as CSI, carrier phase or received power with respect to an array of static reference devices can be used to construct a unique fingerprint which can be mapped to a specific location or region. A database can be constructed of such fingerprints collected across a wide area, and used to construct an estimated channel model of that area.

Up until the last few years, the development of fingerprinting models for indoor positioning has largely treated fingerprints as simple vectors, utilising techniques such as artificial neural networks[6, 7], decision trees[8] and k-Nearest Neighbour[9]. While this format is common in machine learning, positioning is an inherently *geometric* problem; any such model will ultimately infer the spatial relationship that maps fingerprints to coordinates, but 'flattening out' each fingerprint creates difficulty in incorporating information already known about the geometric structure of the environment or the relationship

WIPHAL 2024: Work-in-Progress in Hardware and Software for Location Computation, June 25-27, 2024, Antwerp, Belgium

[*]Corresponding author.

liam.self@cl.cam.ac.uk (L. M. Self); ijw24@cam.ac.uk (I. J. Wassell)

0000-0002-7076-6613 (L. M. Self); 0000-0001-7927-5565 (I. J. Wassell)

between individually collected fingerprints.

More recently, a field has emerged which extends conventional deep learning architectures to accommodate spatial and structural relationships, by operating directly on data structured as graphs[10, 11]. Several models have been proposed since on graph-based positioning, but in general each proposition focuses on a single structuring of the fingerprint database [12]. In this paper, we present a current work-in-progress study which fuses two graph representations of a RF-based fingerprint database to increase the expressiveness of the training data.

The remainder of this paper is organized as follows: In Section 2, we briefly present Graph Neural Networks and the specific architectures used in this study. Section 3 describes the generation of a simulated indoor positioning dataset and the multi-graph model we evaluate. The current results of our work are given in Section 4, and in Section 5 we summarize our work so far and discuss the next stages of our study.

2. Background on Graph Neural Networks

Graph Neural Networks[13] differ fundamentally from conventional neural networks in that the latter can only operate on data ordered in vector or matrix form. GNNs, by comparison, can operate directly on graph-structured data, allowing them to better leverage the relationship between data. We define a graph as a pair $G = (V, E)$ comprising a set of *nodes* — or *vertices* — V and a set of *edges* $E \subseteq V \times V$ defining relations between nodes. A node typically consists of a feature vector of fixed and consistent size (*heterogeneous* graphs that contain nodes of varying feature size do exist but are not considered for the purpose of this paper). GNNs are able to operate on graphs of arbitrary size, and are typically invariant to the ordering of nodes.

2.1. Graph Convolutional Networks

In a conventional feedforward neural network, each layer l applies a transformation

$$h^{(l+1)} = \sigma\left(\boldsymbol{W} h^{(l)} + b\right) \tag{1}$$

where $\boldsymbol{W}$ is a trainable weight matrix and b is a trainable additive bias. σ is a non-linear activation function such as ReLU. We let $h^{(0)}$ be the original input feature vector. This forms the basis for the transformation of node features in a *Graph Convolutional Network* (GCN). However, instead of a node applying a transformation of its own representation from the previous layer, a GCN aggregates the features of the node's neighbours, a process known as *message passing*:

$$h_i^{(l+1)} = \sigma\left(\sum_{v \in \mathcal{N}_i} \frac{1}{\deg(i)} \boldsymbol{W} h_v^{(l)} + b\right) \tag{2}$$

Here, $\mathcal{N}_i$ is the neighbourhood of the ith node, and $h_v^{(l)}$ the node embedding of neighbour v at the lth layer. $\frac{1}{\deg(i)}$ is a normalization constant which applies a weighting by the size of the node's neighbourhood. One can observe an analogy between the aggregation of neighbouring nodes in a GCN and the convolution of a filter with, say, an image within a layer of a Convolutional Neural Network.

By applying Equation 2 over a number of layers, a given node's representation incorporates information from deeper into the network. A three-layer GCN, then, will incorporate the features from nodes three 'hops' from that node. Reaching a balance in the number of layers to use is an important consideration; in a feedforward neural network, too many layers can result in vanishing gradients or overfitting. In a GCN, there is an additional risk of every node embedding essentially converging to the same value, as more nodes are being shared in the aggregations of each node.

Kipf et al.[14] further extended the update function to account for the observation that nodes with a larger neighbourhood will naturally disseminate information wider than one with fewer neighbours.

They update the normalization constant of Equation 2 to redress this by increasing the weighting of nodes with smaller neighbourhoods:

$$h_i^{(l+1)} = \sigma \left(\sum_{v \in \mathcal{N}_i} \frac{1}{\sqrt{\deg(i)}\sqrt{\deg(v)}} \boldsymbol{W} h_v^{(l)} + b \right) \tag{3}$$

In some implementations, edges can be optionally assigned a numerical weight, to increase the influence of certain nodes over others. For example, in a geometric scenario, one could weight edges by similarity or distance between nodes. After a set number of message passing layers, the resulting node embeddings can be returned as-is, further transformed, for example via a feedforward neural network, or aggregated with other nodes to give a single graph-level representation.

2.2. Graph Attention Networks

Graph Attention Networks [15] (GATs) are an adaptation of GCN which differ in the way in which they aggregate each node's neighbourhood. From Equation 3 one can note that the update functions is an isotropic aggregation in which each node equally contributes towards the new representation of a node, while other implementations that allow for edge weighting require these weights to be known and specified in advance. GATs assign a different weighting to each edge via learnable 'attention coefficients':

$$z_i^{(l)} = \boldsymbol{W} h_i^{(l)} \tag{4}$$

$$e_{ik}^{(l)} = \text{LeakyReLU}(a^{(l)^T}(z_i^{(l)} \oplus z_j^{(l)})) \tag{5}$$

$$\alpha_{ij}^{(l)} = \frac{\exp(e_{ik}^{(l)})}{\sum_{k \in N(i)} \exp(e_{ik}^{(l)})} \tag{6}$$

$$h_i^{(l+1)} = \sigma \left(\sum_{j \in \mathcal{N}(i)} \alpha_{ij}^{(l)} W^{(l)} h_i^{(l)} \right) \tag{7}$$

where $\oplus$ is the concatenation operator. $a_{ij}^{(l)}$ applies a softmax to the so-called 'attention score' $e_{ij}^{(l)}$, the purpose of which is to calculate the relative importance of a particular neighbour.

3. Methodology

3.1. Proposed model

We propose a graph-based model that models both the relationship between individual fingerprints in a pre-collected database, and the underlying geometric structure of the environment. To achieve this, we consider a fingerprint database comprising the received power, phase and delay spread between a given location and a set of static reference points (RPs) within an indoor environment. This database is then transformed into two graph representations; one based primarily on the sampled locations, and the other on the reference points.

In the former case, which we will denote as the *fingerprint graph*, we define a graph $G_{\text{fp}}(V_{\text{fp}}, E_{\text{fp}})$ with each node $v_i \in V_{\text{fp}}$ being a point in the fingerprint database sampled at a given location with initial node feature $h_v^{(0)}$ set to the concatenated fingerprint (comprising the received power, phase and delay spread) from each RP at v. In our study we use an environment with eight RPs, and therefore have an initial feature vector of length 24. For each node $v \in V_{\text{fp}}$, for some value $k \in \mathbb{N}, k \geq 1$ an incoming edge is created between the k most similar fingerprints (by euclidean distance) and v. In our experiments we find that $k = 3$ yields optimum performance. This graph is static throughout training,

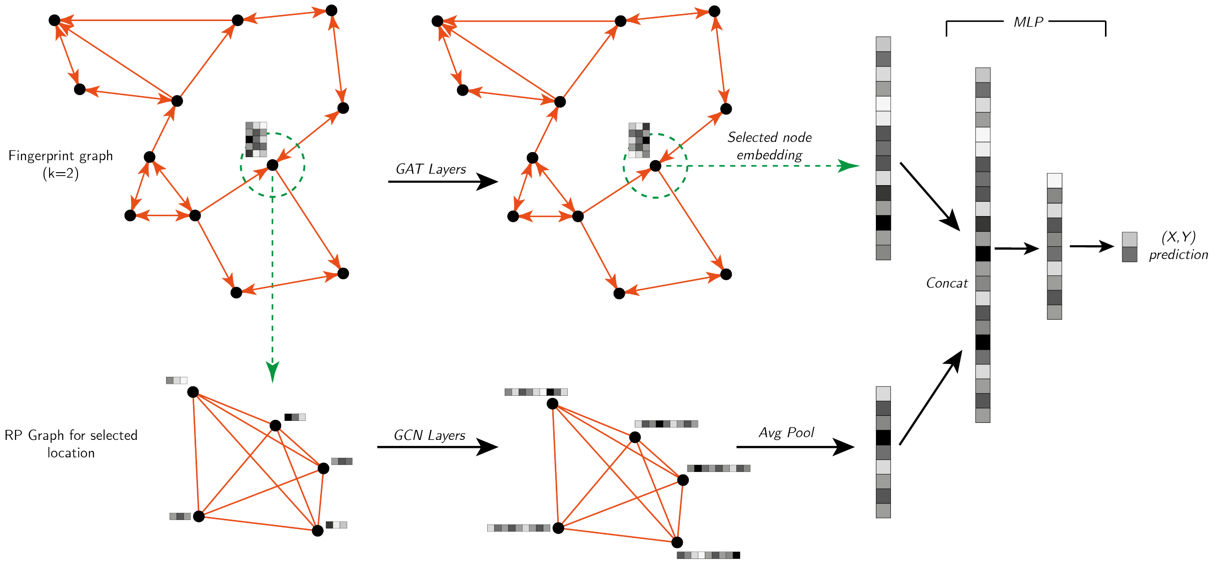


Figure 1: Summary schematic of our proposed model

meaning its edge set and initial nodes states are unchanged. When evaluating an unseen location, the new fingerprint is incorporated into the graph, and the k incoming edges for the new node computed, but the remainder of the graph remains unchanged.

For the *RP graph*, G_{RP}, we instead define a fully connected graph (every node has a bidirectional edge to all other nodes, without, in this case, self-loops) in which the nodes are defined as the individual RPs. The node features for this graph are unique to each individual location sampled in the fingerprint database; for each location we get the initial node state of the 3-feature fingerprint for each RP.

We divide our fingerprint database into a train and test set, and construct a single fingerprint graph for the training set. When considering a given location, we identify that node in the fingerprint graph, and distribute its feature vector across the nodes in our RP graph. We pass the fingerpring graph through a GAT network, and the RP graph through a number of GCN layers. ReLU is used as the non-linearity for both models. The final node embedding of the targeted node is then extracted from the GAT network, and concatenated with a Global Mean Pool[1] readout of the final RP graph. The resulting vector is then passed through a feedforward neural network, finally outputting a length-2 vector which we take as the coordinate prediction. A schematic of our model is shown in Figure 1.

3.2. Data generation

To initially evaluate the potential of our positioning model, we constructed an approximately $368\,\mathrm{m}^2$ simulated environment indicative of a medium-sized office floor, comprising a number of rooms varying from $4.1\,\mathrm{m}^2$ to $50\,\mathrm{m}^2$ connected by corridors. The environment was constructed using the Blender® 3D modelling software[16], and imported into the Remcom Wireless InSite® EM propagation software[17]. The environment was divided into a grid of 3.68 million $1\,\mathrm{cm}^2$ squares, into which a receiver antenna was positioned. Eight transmitters were placed throughout the environment, along the main corridor and in each of the large atriums. Figure 2a shows the positions of the transmitters, indicated in green.

The receivers and transmitters were configured according to the parameters given in Table 1 and the EM environment configured as in Table 2, and pairwise propagation simulations performed between every Tx and Rx, from which received power, phase and delay spread were obtained. An example of a received power plot is shown in Figure 2c.

While such a high resolution dataset would be impractical to gather in reality, we are able to sample from the dataset to emulate measurement campaigns of varying degrees of coverage and uniformity

[1] Global Mean Pool aggregates the information in a graph by taking a mean average of each feature across all nodes in the graph.

Table 1
Antenna Parameters

Antenna type	Omnidirectional
Maximum Gain	$2.5\,\mathrm{dBi}$
Receiver Threshold	$-142\,\mathrm{dBm}$
Carrier Frequency	$2.4\,\mathrm{GHz}$

Table 2
Simulation Parameters

Reflections	6
Transmissions	8
Outer wall material	Brick
Inner wall material	Layered drywall
Floor/Ceiling material	Concrete
InSite Propagation Model	X3D Ray
Ceiling Height	$3\,\mathrm{m}$

Table 3
Multi-graph model hyperparameters

Hyperparameter	Search space	Value in optimum model
Fingerprint GAT final embedding size	24, 48, 96	48
Fingerprint GAT layers	1, 2, 3, 4	2
Fingerprint GAT hidden embedding size	8, 16, 32, 64	32
RP GNN final embedding size	10, 20, 40	20
RP GNN layers	1, 2, 3, 4	2
FC Layers	[64, 32, 16], [32, 16], [16,8]	[32,16]

without the need to perform multiple time-consuming simulations. In this instance, to replicate the performing of a realistic fingerprint measurement campaign as closely as possible, as well as to allow for the future extension of this environment towards training with sequential path data (for which the availability of robust public datasets is limited), random walking paths through the environment were generated using MATLAB's Navigation Toolbox. Positions were sampled every $10\,\mathrm{cm}$, and the power, phase and delay spread estimated by means of linear interpolation. A total of 137 paths, shown in Figure 2b, were generated for the purpose of this study, comprising 30 496 individual locations. Gaussian noise was then applied to the data to add additional uncertainty to the simulated data.

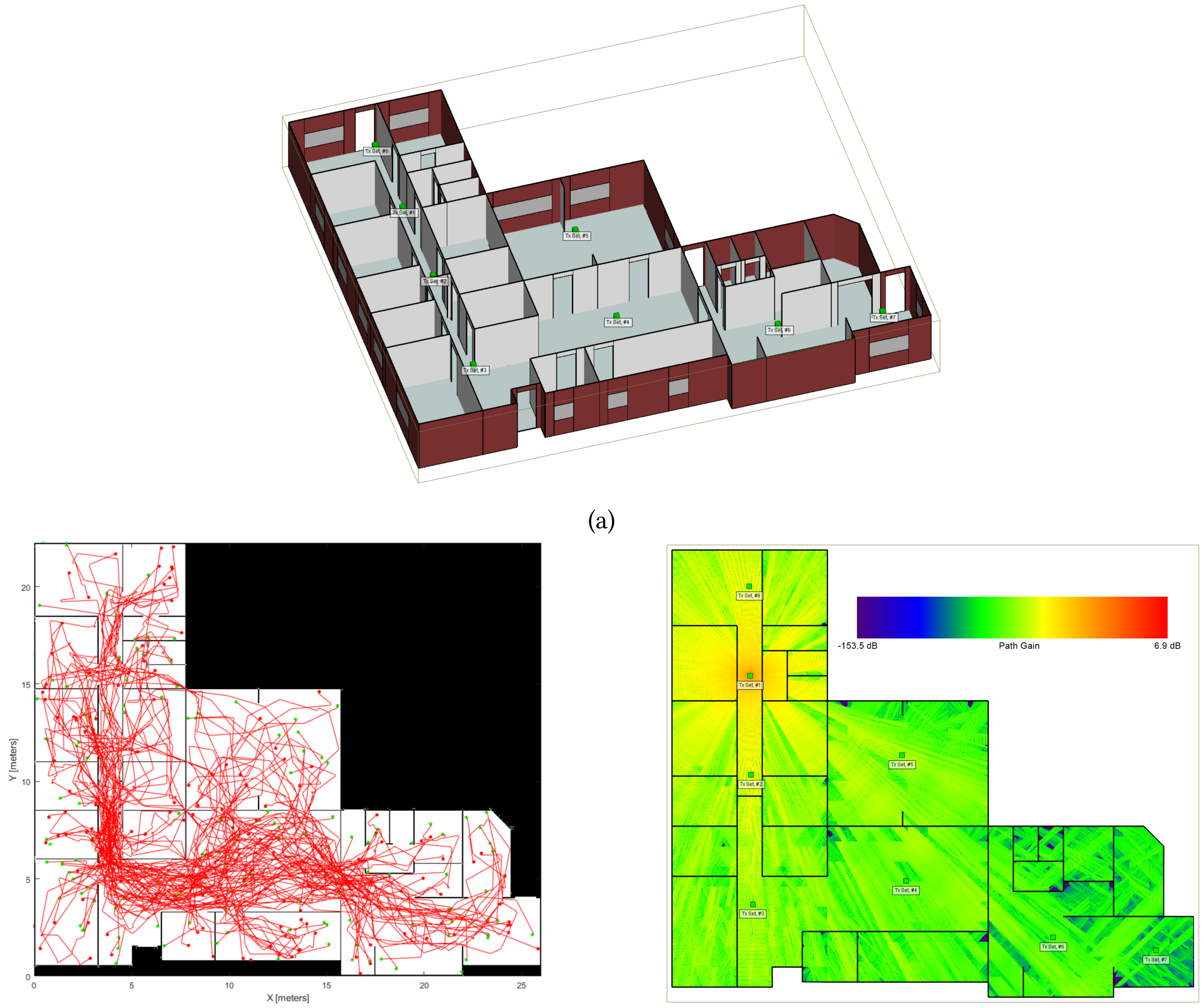
3.3. Pre-processing

All data in the fingerprint database, including the ground truth labels, was standardized prior to use using so-called *robust standardization*, a variant of Z-score standardization more robust against outliers which uses the median and inter-quartile range of each feature in place of mean and standard deviation, respectively.

4. Results

The graph models were implemented using PyTorch Geometric[18], an extension to PyTorch for graph-based learning, and trained using the Adam optimizer, and MSE loss functions. Batch size was set to 32, and the learning rate 1×10^{-4}. The hyperparameters tuned for our model are given in Table 3.

We compared the accuracy of our model against several reference models; kNN, for values of k from 1 to 30 and using cosine, euclidean and Manhattan distance metrics; Random Forest[19] with 8, 16, 32, 64, 128, 256, 512 and 1024 trees; and a multilayer perceptron (MLP), with tuned hyperparameters for



(a)

(b)

(c)

Figure 2: Simulated office environment used for study.
(a) 3D model of the environment showing positions of transmitters (green points);
(b) 'Walking paths' used to generate fingerprint database;
(c) Indicative received power plot at Transmitter 1

the number of layers and hidden channel size. In addition, the multi-graph method was compared with the performance of each respective graph model alone.

The performance of the optimum models and benchmarks are summarized in Figure 3. We can see from the results that our model appears to improve upon these benchmark models by a significant degree, with a 29% reduction in average error than using just an MLP, the best performing non-graph model. It is also interesting to observe that the combination of both the fingerprint and RP graphs results in a greater accuracy than either graph model used individually. This suggests that both graph representations capture and utilise information not present in the other, and further supports the argument that graph-based models are a promising and effective way of incorporating relevant structural information that would otherwise have been lost in a simple vector-based fingerprinting model.

We observe that our model results in significantly fewer outliers, which contributes strongly to the improvement in mean accuracy, though median error is currently higher than our benchmark models (Figure 4. The maximum error encountered using Random Forest, for example, was over 30 metres, compared to $5.43\,\mathrm{m}$ using our own model. The coverage of our dataset is not uniformly distributed throughout the environment; the corners of rooms in particular are sparsely sampled in comparison

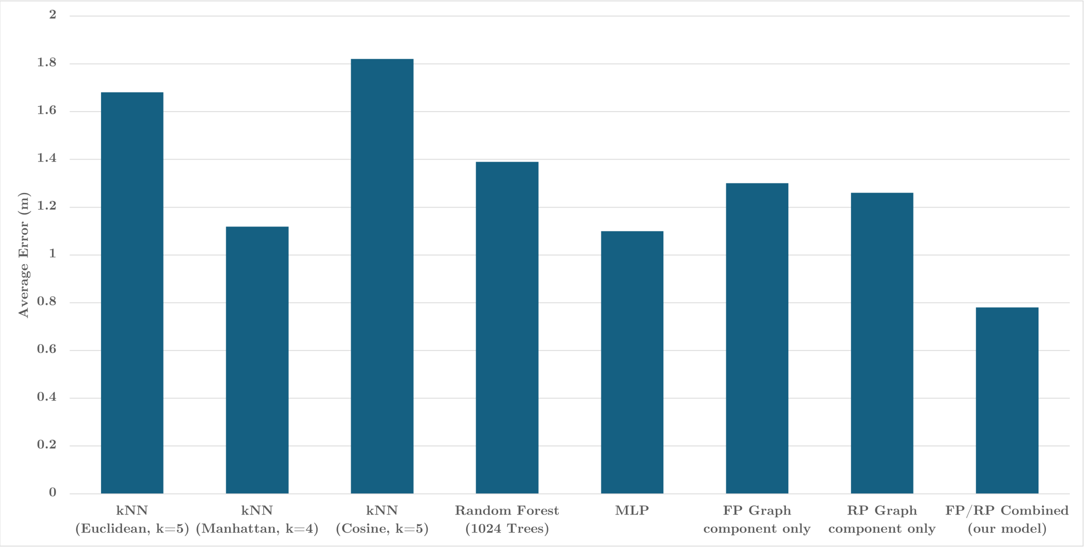


Figure 3: Comparison of our model with reference benchmarks

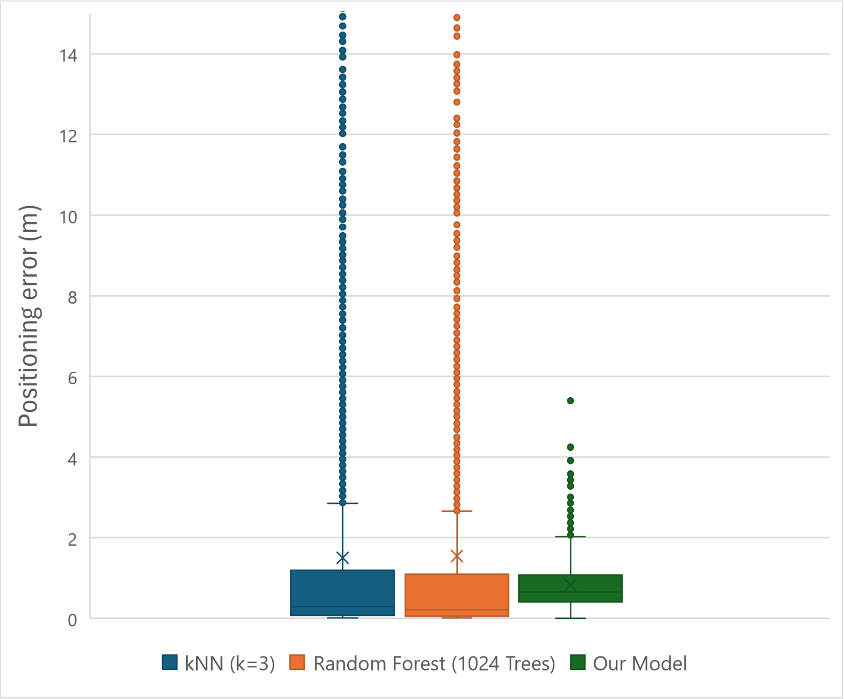


Figure 4: Hair-and-whisker plots for kNN, Random Forest, and our model.

to high-traffic areas such as the main corridors. Distributed and crown sourced walking surveys is a common means of data collection for positioning systems [20], though more so in models that locate a user to within a discrete region as opposed to specific coordinates, and achieving uniformly distributed training data can be time consuming. Accordingly, the kNN model struggled to correctly position locations at the boundaries of the sampled area. By comparison, our model was able to position outlying test locations to within 2 m, even in these sparse regions.

With respect to data collection, we note that the 10cm resolution at which locations were sampled for the purpose of training the models may present challenges when transferring this method to the real world. While this resolution may be possible at an acceptable sampling frequency using bespoke hardware, smartphones are typically constrained in their maximum sample rate. In recent versions of the Android operating system, for example, the time to complete a scan of broadcasting WiFi devices and their signal strengths is around 3-5 s depending on handset model[21]. A data collection campaign of the scale described in our paper could take many hours to complete, even accounting for a division of labour using multiple devices. A possible solution to this is currently ongoing, which is investigating the feasibility of transfer learning, which has previously been applied to RF data[22, 23], to augment a smaller amount of manually sampled locations with a higher resolution synthetic dataset, to allow for a shorter data collection campaign while maintaining acceptable localisation performance.

5. Conclusion and future work

We have presented a new RF-based positioning model which combines two different graph representations of a simulated indoor environment. Current results have indicated a 32 cm improvement over all benchmarks, and stronger performance compared to the use of only one or the other graph representation. Considering future work, with respect to the fingerprint graph, at present we only consider each location independently, and incorporating new ones into the graph does not affect the rest of the graph (new incoming edges can added to allow information to propagate to unseen nodes, but this does not allow information flow in the other direction). One might consider the case where multiple users are present in the same environment. In this case, it may be advantageous to consider the relationship between users, turning this problem into one of collaborative positioning. Future work is, at the time of writing, underway to investigate the potential for this model to be extended into a collaborative one, to explore the use of temporal data (the dataset having been constructed as a set of walking paths) to further improve model performance, and to further evaluate performance against industry standard datasets such as UJIIndoorLoc[24]. Noting the improvement in accuracy in outlying regions of our environment, we also wish to explore the extent to which our model is able to mitigate against inconsistent and non-uniform measurement campaigns.

Acknowledgments

Liam M. Self was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) grant EP/S023046/1 for the EPSRC Centre for Doctoral Training in Sensor Technologies for a Healthy and Sustainable Future.

References

[1] A. H. Ismail, H. Kitagawa, R. Tasaki, K. Terashima, Wifi rss fingerprint database construction for mobile robot indoor positioning system, in: 2016 IEEE international conference on systems, man, and cybernetics (SMC), IEEE, 2016, pp. 001561–001566.

[2] P. Octaviani, W. Ce, Inventory placement mapping using bluetooth low energy beacon technology for warehouses, in: 2020 International Conference on Information Management and Technology (ICIMTech), 2020, pp. 354–359. doi:10.1109/ICIMTech50083.2020.9211206.

[3] I. Rubino, J. Xhembulla, A. Martina, A. Bottino, G. Malnati, Musa: Using indoor positioning and navigation to enhance cultural experiences in a museum, Sensors 13 (2013) 17445–17471.

[4] Y. Liu, X. Fang, F. Lu, X. Chen, X. Huang, Indoor positioning and prediction in smart elderly care: Model, system and applications, in: M. Qiu (Ed.), Algorithms and Architectures for Parallel Processing, Springer International Publishing, Cham, 2020, pp. 537–548.

[5] T. Tegou, I. Kalamaras, M. Tsipouras, N. Giannakeas, K. Votis, D. Tzovaras, A low-cost indoor activity monitoring system for detecting frailty in older adults, Sensors 19 (2019). URL: https://www.mdpi.com/1424-8220/19/3/452. doi:10.3390/s19030452.

[6] M. Ibrahim, M. Torki, M. ElNainay, Cnn based indoor localization using rss time-series, in: 2018 IEEE Symposium on Computers and Communications (ISCC), 2018, pp. 01044–01049. doi:10.1109/ISCC.2018.8538530.

[7] W. Shao, H. Luo, F. Zhao, Y. Ma, Z. Zhao, A. Crivello, Indoor positioning based on fingerprint-image and deep learning, IEEE Access 6 (2018) 74699–74712. doi:10.1109/ACCESS.2018.2884193.

[8] D. Sánchez-Rodríguez, P. Hernández-Morera, J. M. Quinteiro, I. Alonso-González, A low complexity system based on multiple weighted decision trees for indoor localization, Sensors 15 (2015) 14809–14829. URL: https://www.mdpi.com/1424-8220/15/6/14809. doi:10.3390/s150614809.

[9] P. Torteeka, X. Chundi, Indoor positioning based on wi-fi fingerprint technique using fuzzy k-nearest neighbor, in: Proceedings of 2014 11th International Bhurban Conference on Applied Sciences & Technology (IBCAST) Islamabad, Pakistan, 14th - 18th January, 2014, 2014, pp. 461–465. doi:10.1109/IBCAST.2014.6778188.

[10] W. Yan, D. Jin, Z. Lin, F. Yin, Graph neural network for large-scale network localization, in: ICASSP 2021 - 2021 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), 2021, pp. 5250–5254. doi:10.1109/ICASSP39728.2021.9414520.

[11] X. Kang, X. Liang, T. Wang, Indoor localization algorithm based on geometric deep learning, in: Z. Zhang, C. Li (Eds.), Fifteenth International Conference on Signal Processing Systems (ICSPS 2023), volume 13091, International Society for Optics and Photonics, SPIE, 2024, p. 1309121. URL: https://doi.org/10.1117/12.3025056. doi:10.1117/12.3025056.

[12] F. Lezama, G. G. Gonzalez, F. Larroca, G. Capdehourat, Indoor Localization using Graph Neural Networks, 2021 IEEE URUCON, URUCON 2021 (2021) 51–54. doi:10.1109/URUCON53396.2021.9647082.

[13] F. Scarselli, M. Gori, A. C. Tsoi, M. Hagenbuchner, G. Monfardini, The graph neural network model, IEEE Transactions on Neural Networks 20 (2009) 61–80. doi:10.1109/TNN.2008.2005605.

[14] T. N. Kipf, M. Welling, Semi-Supervised Classification with Graph Convolutional Networks, 5th International Conference on Learning Representations, ICLR 2017 - Conference Track Proceedings (2016). arXiv:1609.02907.

[15] P. Veličković, A. Casanova, P. Liò, G. Cucurull, A. Romero, Y. Bengio, Graph Attention Networks, 6th International Conference on Learning Representations, ICLR 2018 - Conference Track Proceedings (2017). doi:10.1007/978-3-031-01587-8_7. arXiv:1710.10903.

[16] B. O. Community, Blender - a 3D modelling and rendering package, Blender Foundation, Stichting Blender Foundation, Amsterdam, 2018. URL: http://www.blender.org.

[17] Wireless InSite 3D Wireless Propagation Software, 2023. URL: https://www.remcom.com/wireless-insite-propagation-software, accessed: 2024-04-06.

[18] M. Fey, J. E. Lenssen, Fast graph representation learning with pytorch geometric, arXiv preprint arXiv:1903.02428 (2019).

[19] T. K. Ho, Random decision forests, in: Proceedings of 3rd international conference on document analysis and recognition, volume 1, IEEE, 1995, pp. 278–282.

[20] S. H. Jung, B.-C. Moon, D. Han, Performance evaluation of radio map construction methods for wi-fi positioning systems, IEEE Transactions on Intelligent Transportation Systems 18 (2017) 880–889. doi:10.1109/TITS.2016.2594479.

[21] H.-H. Liu, C. Liu, Implementation of wi-fi signal sampling on an android smartphone for indoor positioning systems, Sensors 18 (2017) 3.

[22] S. Kuzdeba, J. Robinson, J. Carmack, Transfer learning with radio frequency signals, in: 2021 IEEE 18th Annual Consumer Communications & Networking Conference (CCNC), 2021, pp. 1–9. doi:10.1109/CCNC49032.2021.9369550.

[23] M. Iwasaki, T. Nishio, M. Morikura, K. Yamamoto, Transfer learning-based received power prediction with ray-tracing simulation and small amount of measurement data, in: 2020 IEEE 92nd Vehicular Technology Conference (VTC2020-Fall), IEEE, 2020, pp. 1–6.

[24] J. Torres-Sospedra, R. Montoliu, A. Martínez-Usó, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, J. Huerta, Ujiindoorloc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems, in: 2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN), 2014, pp. 261–270. doi:10.1109/IPIN.2014.7275492.