Efficient and Robust WiFi Indoor Positioning using Hierarchical Navigable Small World Graphs

Max Willian Soares Lima[1,2], Horacio A. B. Fernandes de Oliveira[1], Eulanda Miranda dos Santos[1], Edleno Silva de Moura[1], Rafael Kohler Costa[3], Marco Levorato[4]

[1]Institute of Computing, Federal University of Amazonas, Brazil
{mw_ac, horacio, emsantos, edleno}@icomp.ufam.edu.br
[2]Institute of Innovation, Research, and Scientific Development of Amazonas, Brazil
[3]Education Technologies, Positivo Technologies, Brazil
rkcosta@positivo.com.br
[4]Computer Science Department, University of California, Irvine, USA
levorato@uci.edu

***Abstract*—Indoor positioning systems consist of identifying the physical location of devices inside buildings. They are usually based on the signal strength of a device packet received by a set of WiFi access points. Among the most precise solutions, are those based on machine learning algorithms, such as kNN (k-Nearest Neighbors). This technique is known as fingerprint positioning. Even though kNN is one of the most used classification methods due to its high precision results, it lacks scalability since an instance we need to classify must be compared to all other instances in the training base. In this work, we use a novel hierarchical navigable small world graph technique to fit the training database so that the samples can be efficiently classified in the online phase of the fingerprint positioning, allowing it to be used in large-scale scenarios and/or to be executed in resource-limited devices. We evaluated the performance of this solution using both synthetic and real-world training data and compared its performance to other known kNN variants such as kd-tree and ball-tree. Our results clearly show the performance gains of the graph-based solution, while still being able to maintain or even reduce the positioning error.**

***Index Terms*—indoor positioning systems, k-nearest neighbors, hierarchical small world graphs**

I. Introduction

Positioning systems allow the estimation of the physical location (i.e., latitude, longitude) of devices [1]–[3]. The use of Location Based Services (LBSs) in outdoor environments (e.g., Waze, Google Maps) increased a lot in recent years due to the availability of GPS (Global Positioning System) in most smartphones [4]. However, the same cannot be said about LBSs in *indoor* environments such as buildings or parking lots since the lack of direct visibility to the GPS satellites results in unacceptable errors.

With the growing number of WiFi devices (e.g., smartphones, smart TVs, Internet of Things), the viability of using existing WiFi infrastructure as a platform for LBSs is becoming increasingly attractive. In such cases, one of the key points is to provide precise, scalable, and real-time positioning information about the mobile devices, which is the main goal of Indoor Positioning Systems (IPSs).

Among the most viable solutions, are those based on supervised machine learning [5]–[7], that use previously recorded received signal strengths to train a classifier, such as kNN (k-Nearest Neighbors) [7]–[9], so it can be used to estimate the nodes positions later in an online phase. This technique is known as fingerprint positioning and it has been used mainly due to two reasons: (1) the availability of signal strength information without requiring extra hardware; and (2) the system's immunity to the unpredictability of signal strength values in indoor environments. Several improvements to the fingerprint positioning have been proposed in order to reduce its positioning error so that in some works it is already possible to localize devices in an indoor environment with errors of only a meter or so [5], [8], [10].

Even though kNN is one of the most used classification methods due to its high precision results [11], it lacks scalability since an instance we need to classify must be compared to all other instances in the training database [12]. Also, this training database tends to drastically increase in size as we increase the location scenario or when we need more precision, since the more data we have, the more precise will be the estimated positions [13].

In this work, we use a novel hierarchical navigable small world graph technique to fit the training database so that samples can be efficiently classified in the online phase of the fingerprint positioning, allowing it to be used in large-scale scenarios and/or to be executed in resource-limited devices. To the best of our knowledge, this is the first time a graph-based classification algorithm is applied to IPSs. We evaluated the performance of our solution using synthetic, large-scale fingerprint positioning data and also real-world training data [14] and compared its performance to other known kNN optimizations such as kd-tree and ball-tree.

The remainder of this paper is organized as follows. In the next section, we present our related work. The graph-based technique is then explained in section III. We show and discuss our performance evaluation in Section IV. Finally, in section V we present our conclusions and future work.

II. Related Work

Most indoor positioning systems solutions have focused on reducing the localization error [7], [13]. For instance, RADAR [15] is one of the first works to propose the use of signal strength data with machine learning in order to localize nodes inside a building [16], since traditional multilateration-based localization performed poorly in these environments.

The high computational cost of IPSs has also been addressed in the literature [11]. A joint clustering technique has been proposed by Youssef et al. [16] that groups locations based on their set of WiFi access points in order to reduce the computational cost. Another technique to reduce the size of the training database is proposed by Chen et al. [13], in which they selectively choose a subset of the access points for the purpose of clustering and then use a decision-tree-based model for the position estimation. Fang et al. [5] also reduce the dimensionality of the problem by combining related access point information into principal components, reducing the final computational cost of the system. Finally, in order to reduce the computational complexity, Yim et al. [7] proposed the discretization of the signal strength values (e.g., values between -40 and -30dBm are transformed to "I2") and then a decision-tree-based mechanism is created and used for the position estimation.

In our work, we are mainly focusing on solutions that do not lose training information in order to achieve better performance. Following this path, some well-known classification algorithms have been proposed such as ball-tree [17] and kd-tree [18]. These and other approaches [7], [13] are based on a tree structure in order to reach and find quickly the area of the training database with similar attributes. After that, the k nearest neighbors can easily be found. More recently, the use of graph-based approaches has shown competitive performances results on data sets with high dimensions, which is the case of the IPSs, as seen in [19] and [20]. In this work, we go further and experiment with the use of hierarchical navigable small world graphs, as recently proposed by Malkov et al. [21], in order to reduce the computational cost of IPSs.

III. Indoor Positioning using Hierarchical Navigable Small World Graphs

Recently, some works have shown significant advances in the use of graph-based methods to search for the nearest neighbors on high dimensional datasets. This is an important aspect for fingerprint-based high-scale indoor positioning systems since a training database can be composed of hundreds to millions of access points (i.e., attributes, dimensions).

These graph-based structures are commonly used with a small world model, which has two important features: small values of path length among the nodes and high values of clustering coefficient [22]. The first feature allows the fast search of a similar instance in the training database. Basically, a small percentage of distant nodes will have links among them, allowing the search to bypass most of the database in a single hop. The second feature guarantees the connectivity of the nodes and makes it easy to find the k nearest values of a determined instance. Following this path, a proximity graph for nearest neighbor search called Navigable Small World (NSW) was proposed in [23] and [24].

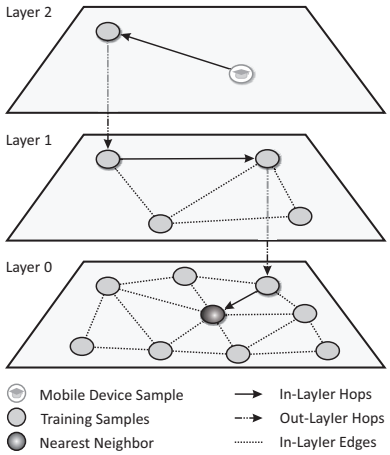


Fig. 1: Indoor positioning system classification using hierarchical navigable small world graphs, based on [21].

The Hierarchical Navigable Small World Graphs (HNSW), proposed in [21], implemented the idea of representing each instance of the dataset as a node in the graph. It separates their links according to their length scale, producing a multilayer graph that allows the evaluation of only a small subset of the connections for each element, thus reducing the computational cost of performing the search.

To build the multilayer graph, the links are set with different distance scales by artificially introducing layers, as depicted in Figure 1. Each layer is represented by an integer number, called level number, and every node on the graph receives a level number representing the maximum layer for which this node belongs to. The top layers represent connections between distant nodes, which gives the graph its main small world characteristics. On the bottom layers, the graph will have the connections between closer nodes, connecting a node to its neighborhood. Each node is inserted on the graph at the layer represented by its level number, as well as at the descending layers up until the ground layer (Layer 0), being linked to its nearest nodes in each level. The level number is randomly selected and an exponentially decaying probability distribution is used in order to concentrate the nodes at the lower layers and enable close connections, and letting only a few nodes at the top layers, using a parameter to normalize this distribution.

As depicted in Figure 1, after the graph construction, when a new search is required, it will start on the highest layer and will greedily traverse the graph selecting elements only at that layer, until it finds a local optimum. Once the local optimum is found, the routing will continue on the next layer below in the hierarchy. This process will be repeated starting from the local minimum found in the previous layer, using its nodes as entry points, and so on until the ground layer is reached.

In this work, we implemented the hierarchical navigable small world graphs in order to evaluate its performance in indoor positioning systems. Our methodology and the obtained results are discussed in the next section.

IV. Performance Evaluation

Our performance evaluation aimed at analyzing the time complexity of building the search index (model fitting) and also to find the nearest neighbors (instance classification). In addition, we analyzed the resulting positioning error of the indoor positioning system. We compared the hierarchical navigable small world graphs (HNSW) to two other tree-based methods: kd-tree and ball-tree. We also compared the results to the classic kNN search, commonly known as brute force. For the implementation of the HNSW search, we used the Non-Metric Space Library (NMSLIB) [25], while for the other methods we used the Scikit-Learn [26], both using Python language.

We evaluated the performance of the implemented solutions using both synthetic and real-world training data. The synthetic training data was adopted to generate large-scale training data composed by hundreds of access points and millions of training points, allowing us to experiment with the methods in controlled scenarios. In order to create these synthetic scenarios, we also implemented a signal strength simulator based on the known simple indoor signal propagation model [27]. This model takes into consideration both wall and floor attenuation factors, as discussed in [15].

Finally, as for the real-world scenario training data, we used the popular UJI Indoor Localization dataset [14]. Just as a comparison, the UJI dataset, which is the largest real-world indoor positioning dataset found in the literature, contains little more than 20 thousand instances, while our generated synthetic data contains almost 270 thousand instances. All of the results depicted in the graphs were reached through the mean of a ten-fold cross-validation [28].

A. Impact of the Number of Instances on the Classification

The main goal of using the HNSW technique is to decrease the classification time while maintaining or decreasing the positioning error. Thus, in our first experiment, we varied the number of instances in the training database from 26 thousand to 269 thousand instances. In this scenario, we had 196 access points (attributes in the dataset).

Figure 5(a) compares the time required for the experimented methods to classify 30% of the dataset (used as the test dataset) using the synthetic scenario. As we can see, the difference among all optimized methods to the kNN using brute force is considerable. The graph-based solution is 96% faster than the classic brute force kNN, while the Kd-Tree and Ball-Tree was 85% and 77% faster, respectively.

Since the difference from the optimized methods to the brute force is high, and in order to facilitate the visualization, Figure 5(b) repeats the previous experiment without executing the brute force. Now, we can clearly see that the HNSW technique is also able to outperform the tree-based approaches. The obtained results show that the graph-based technique is 59% faster than Kd-Tree and 76% faster than Ball-Tree. For the next experiments, we also decided to remove the brute force.

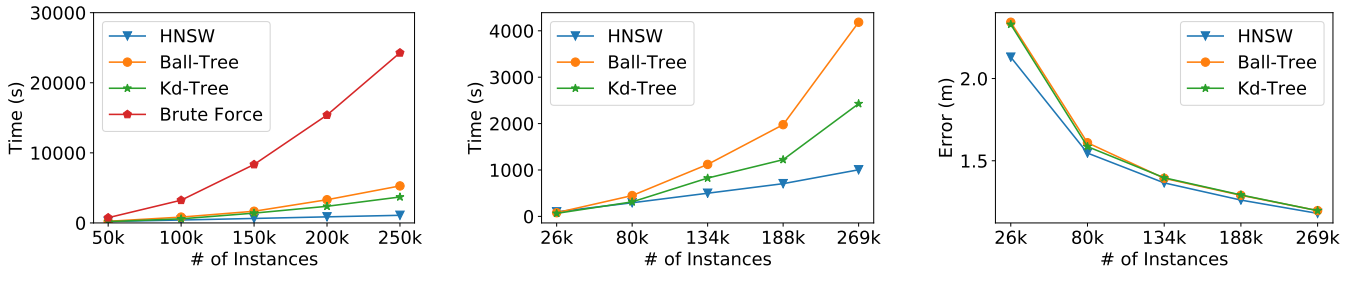
Finally, Figure 5(c) shows the behavior of the positioning error obtained by the methods as we increase the number of instances. As we can see, the positioning error decreases when we have more training data, which can be confirmed by several works from the literature. In our case, the positioning error decreased from 2.33m to 1.19m, a reduction of 49%, as we increase the number of instances from 26 thousand to 269 thousand.

B. Impact of the Number of Instances on the Model Fitting

As mentioned earlier, one of the steps in an indoor positioning system architecture is the model creation (fitting) based on the training data. Even though this time is not that important in IPSs since it is executed only once, it might affect some known techniques that require a periodic model fit, such as adaptive and calibration-based solutions [29] that are known to reduce even more the positioning error over time.

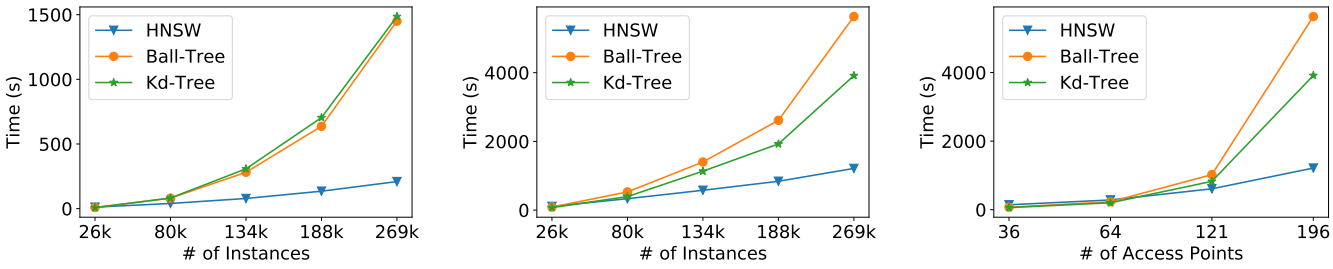
Thus, Figure 6(a) shows the time required to create the machine learning model from a training dataset as it increases in size. The classic kNN (brute force) has a constant near zero time, since it does not really create any model, and thus is not in this graph. As we can see, the creation of the graph-based model is faster than the tree-based models, especially in the larger scenarios where it was 86% faster.

Finally, Figure 6(b) shows the complete total time (fitting+classification) of the experimented models. When including both model creation and instance classification, the HNSW



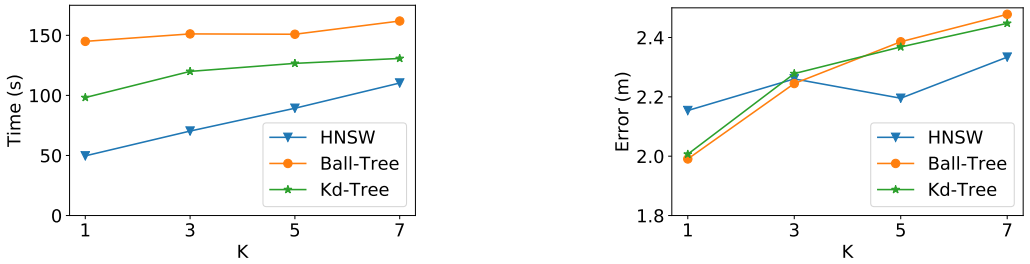
(a) (b) (c)

Fig. 5: Scalability test results from a dataset containing 196 access points (attributes) and varying the number of instances. (a) classification time including the brute force and (b) without the brute force method. (c) positioning error obtained by the experimented methods.



(a) (b) (c)

Fig. 6: Scalability tests including the (a) model fitting time and (b) the total time (fit+classification). (c) impact of the number of access points.



(a) (b)

Fig. 7: (a) time and (b) positioning error results from the real-world scenario dataset when varying the k value.

technique is 79% faster then Ball-Tree and 69% faster than Kd-Tree.

C. Impact of the Number of Access Points

Another factor that greatly impacts the computational cost of an indoor positioning system is the number of access points since a single additional access point will add a new column (attribute) for all of the instances in the training database. Increasing the density of the access points (number of access points per squared meter) is also a known technique used in the literature to reduce the localization error.

In order to evaluate the impact of the number of access points in the computational cost of the experimented methods, we increased this number from 36 to 196 access points. Since we didn't want the error positioning error to be affected, we kept the access points density constant, in such a way that as we increase the number of access points, we also increased the scenario size (number of rooms and number of training points).

As depicted in Figure 6(c), another interesting aspect of the HNSW technique is that its performance advantages also scale up as we increase the number of access points, being 77% faster than Ball-Tree for 196 access points.

D. Real-World Scenario Experiments

Finally, in order to evaluate the performance of the experimented methods using real-world scenario data, we executed our experiments using the popular UJI Indoor Localization dataset [14]. One interesting aspect of this training database is that even though it has 529 access points (attributes), it only has little more than 20 thousand instances, since for each training point there was little to no repetition, i.e., only a single packet was sent by the mobile device from each determined location.

Thus, the UJI training database has only 10% the number of instances of our synthetic database but has, however, 2.7 times the number of access points, which is an interesting and very different scenario. Since we had not much control over the dataset, in order to execute the experiments, we kept all of the scenario characteristics fixed and changed the kNN k value from 1 to 7.

As depicted in Figure7(a), the HNSW method was faster in all of the cases, being 66% faster than Ball-Tree and 50% faster than Kd-Tree for k equal to 1. Analyzing these results, we can also note that the total time for HNSW does increase at a higher rate than the other tree-based methods as we increase the k value, since this k value influences the graph data structure creation.

However, as depicted in Figure 7(b), the lowest positioning errors are obtained by the lowest k values, which can be expected since in this dataset there was almost no repetition of packets from the same location.

V. Conclusions

As Indoor Positioning Systems become more popular, new and novel applications will finally be able to be developed taking advantage of a real-time, precise position information in indoor environments. However, in order for these IPSs to be used in large-scale and real-time scenarios, efficient and robust machine learning algorithms are required.

In this paper, based on some recent works that have shown significant advances in the use of graph-based methods to search for the nearest neighbors on high dimensional datasets, we have proposed the use of hierarchical navigable small world graphs as a machine learning algorithm in high-scale, real-time indoor positioning systems as an alternative to the traditional brute force kNN and also to the classic tree-based optimizations.

Our results clearly show the performance gains of the graph-based solution since it was able to estimate the position of the nodes 96% faster than the classic brute force kNN and at least 77% faster than the tree-based optimizations. All of these while still being able to maintain or even reduce the positioning error in all of the experimented synthetic and real-world scenario datasets.

These results are very promising, but some advantages still need to be further exploited in future works. For instance, we intend to use the knowledge of the last known position of the mobile device, a data easily available in IPSs, to reach even faster the ground layer of the hierarchical navigable small world graph.

References

[1] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of wireless indoor positioning techniques and systems," *IEEE Transactions on Systems, Man, and Cybernetics, Part C (Applications and Reviews)*, vol. 37, no. 6, pp. 1067–1080, Nov 2007.

[2] "Recent Advances in Wireless Indoor Localization Techniques and System."

[3] A. Boukerche, H. A. B. F. Oliveira, E. F. Nakamura, and A. A. F. Loureiro, "Localization systems for wireless sensor networks," *IEEE Wireless Communications*, vol. 14, no. 6, pp. 6–12, December 2007.

[4] A. Boukerche, H. A. Oliveira, E. F. Nakamura, and A. A. Loureiro, "Vehicular ad hoc networks: A new challenge for localization-based systems," *Computer Communications*, vol. 31, no. 12, pp. 2838 – 2849, 2008, mobility Protocols for ITS/VANET.

[5] S. H. Fang and T. Lin, "Principal component localization in indoor wlan environments," *IEEE Transactions on Mobile Computing*, vol. 11, no. 1, pp. 100–110, Jan 2012.

[6] A. H. Sayed, A. Tarighat, and N. Khajehnouri, "Network-based wireless location: challenges faced in developing techniques for accurate wireless location information," *IEEE Signal Processing Magazine*, vol. 22, no. 4, pp. 24–40, July 2005.

[7] "Introducing a decision tree-based indoor positioning technique," *Expert Systems with Applications*, vol. 34, no. 2, pp. 1296 – 1302, 2008.

[8] S. Khodayari, M. Maleki, and E. Hamedi, "A rss-based fingerprinting method for positioning based on historical data," in *Performance Evaluation of Computer and Telecommunication Systems (SPECTS), 2010 International Symposium on*, July 2010, pp. 306–310.

[9] X. Tao, X. Li, J. Ma, and J. Lu, "Cluster filtered knn: A wlan-based indoor positioning scheme," in *2008 International Symposium on a World of Wireless, Mobile and Multimedia Networks(WOWMOM)*, vol. 00, 2008, pp. 1–8.

[10] B. Li, J. Salter, A. G. Dempster, and C. Rizos, "Indoor positioning techniques based on wireless LAN," in *Lan, First Ieee International Conference on Wireless Broadband and Ultra Wideband Communications*, 2006, pp. 13–16, 00402.

[11] T.-N. Lin and P.-C. Lin, "Performance comparison of indoor positioning techniques based on location fingerprinting in wireless networks," in *2005 International Conference on Wireless Networks, Communications and Mobile Computing*, vol. 2, June 2005, pp. 1569–1574 vol.2.

[12] S. P. Kuo and Y. C. Tseng, "Discriminant minimization search for large-scale rf-based localization systems," *IEEE Transactions on Mobile Computing*, vol. 10, no. 2, pp. 291–304, Feb 2011.

[13] Y. Chen, Q. Yang, J. Yin, and X. Chai, "Power-efficient access-point selection for indoor location estimation," *IEEE Transactions on Knowledge and Data Engineering*, vol. 18, no. 7, pp. 877–888, July 2006.

[14] J. Torres-Sospedra, R. Montoliu, A. Martnez-Us, J. P. Avariento, T. J. Arnau, M. Benedito-Bordonau, and 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)*, Oct 2014, pp. 261–270.

[15] P. Bahl and V. N. Padmanabhan, "Radar: an in-building rf-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000. Conference on Computer Communications. Nineteenth Annual Joint Conference of the IEEE Computer and Communications Societies (Cat. No.00CH37064)*, vol. 2, 2000, pp. 775–784 vol.2.

[16] M. A. Youssef, A. Agrawala, and A. U. Shankar, "Wlan location determination via clustering and probability distributions," in *Proceedings of the First IEEE International Conference on Pervasive Computing and Communications, 2003. (PerCom 2003).*, March 2003, pp. 143–150.

[17] S. M. Omohundro, *Five balltree construction algorithms*. International Computer Science Institute Berkeley, 1989, no. ICSI Technical Report TR-89-063.

[18] J. L. Bentley, "Multidimensional binary search trees used for associative searching," *Commun. ACM*, vol. 18, no. 9, pp. 509–517, Sep. 1975.

[19] W. Dong, C. Moses, and K. Li, "Efficient k-nearest neighbor graph construction for generic similarity measures," in *Proceedings of the 20th international conference on World wide web*. ACM, 2011, pp. 577–586.

[20] Y. Malkov, A. Ponomarenko, A. Logvinov, and V. Krylov, "Approximate nearest neighbor algorithm based on navigable small world graphs," *Information Systems*, vol. 45, pp. 61–68, 2014.

[21] Y. A. Malkov and D. A. Yashunin, "Efficient and robust approximate nearest neighbor search using Hierarchical Navigable Small World graphs," *ArXiv e-prints*, Mar. 2016.

[22] D. L. Guidoni, A. Boukerche, L. A. Villas, F. S. de Souza, H. A. Oliveira, and A. A. Loureiro, "A small world approach for scalable and resilient position estimation algorithms for wireless sensor networks," in *Proceedings of the 10th ACM International Symposium on Mobility Management and Wireless Access*, ser. MobiWac '12, 2012, pp. 71–78.

[23] A. Ponomarenko, Y. Malkov, A. Logvinov, and V. Krylov, "Approximate nearest neighbor search small world approach," in *International Conference on Information and Communication Technologies & Applications*, 2011.

[24] Y. Malkov, A. Ponomarenko, A. Logvinov, and V. Krylov, "Approximate nearest neighbor algorithm based on navigable small world graphs," *Information Systems*, vol. 45, pp. 61–68, 2014.

[25] L. Boytsov and B. Naidan, "Engineering efficient and effective non-metric space library," in *Similarity Search and Applications - 6th International Conference, SISAP 2013, A Coruña, Spain, October 2-4, 2013, Proceedings*, 2013, pp. 280–293.

[26] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and E. Duchesnay, "Scikit-learn: Machine learning in Python," *Journal of Machine Learning Research*, vol. 12, pp. 2825–2830, 2011.

[27] D. Plets, W. Joseph, K. Vanhecke, E. Tanghe, and L. Martens, "Simple indoor path loss prediction algorithm and validation in living lab setting," *Wireless Personal Communications*, vol. 68, no. 3, pp. 535–552, Feb 2013.

[28] R. Kohavi *et al.*, "A study of cross-validation and bootstrap for accuracy estimation and model selection," in *Ijcai*, vol. 14, no. 2. Montreal, Canada, 1995, pp. 1137–1145.

[29] Y.-C. Chen, J.-R. Chiang, H.-h. Chu, P. Huang, and A. W. Tsui, "Sensor-assisted wi-fi indoor location system for adapting to environmental dynamics," in *Proceedings of the 8th ACM International Symposium on Modeling, Analysis and Simulation of Wireless and Mobile Systems*, ser. MSWiM '05, 2005, pp. 118–125.