Performance Analysis Of Out-of-the-shelf Regressors for Accurate Indoor Positioning

Bismark C. Teixeira, Julia B. Silva, Diego A. Sousa, Daniel C. Araújo

***Abstract*— This paper presents a novel indoor positioning system utilizing machine learning for user localization based on path loss estimates from strategically placed access points. We compare the performance of different out-of-the-shelf regressors and leverage the Extra Trees Regressor algorithm's randomness, avoiding overfitting and outperforming traditional methods like K-Nearest Neighbors. Our system, simulated under the 3GPP 28 GHz indoor channel model, focuses on improving Root Mean Square Error and R-squared metrics. Findings affirm the system's robustness and machine learning's potential in enhancing indoor positioning accuracy.**

***Keywords*— indoor, machine learning, positioning, tree regressor, R2, MAE**

I. Introduction

Accurate real-time positioning is crucial for enabling location-based service (LBS) and is increasingly important in complex indoor environments due to the rapid expansion of the Internet of Things, advancements in communication technology, and the emergence of Industry 4.0, which, emphasizes the integration of intelligent systems and automation. This showcases the need for precise indoor localization. While outdoor localization benefits from mature technologies like Global positioning service (GPS), indoor spaces present unique challenges due to their complexity and GPS signal obstruction from objects with diverse shapes, and sizes, both stationary and moving, impacts indoor positioning accuracy [1].

To address these challenges and accommodate to the diverse demands of indoor localization applications within Industry 4.0 settings, such as smart factories and warehouses, indoor localization systems must improve system accuracy, power consumption, system magnitude, and deployment efficiency. Developing and integrating various solutions is essential to effectively serve a wide range of indoor localization applications, including determining the location of cars or their owners in underground parking lots [1].

LBS have become integral to modern society, with applications spanning from navigation and emergency response to geotargeted advertising and social networking. To provide accurate positioning data for these services, various ranging techniques are employed, such as Received Signal Strength Indicator (RSSI) [2], Time-of-Arrival (ToA), Time-Difference-of-Arrival (TDoA) [3], and Channel Estimation [4], [5]. RSSI calculates the distance between devices by measuring the power of received signals, which attenuate over distance. ToA measures the time it takes for a signal to travel from the transmitter to the receiver, while TDoA compares the arrival times of signals at multiple receivers to triangulate a location. Channel Estimation, on the other hand, involves the estimation of the characteristics of the transmission channel to refine the accuracy of positioning. By leveraging these techniques, LBS providers can accommodate to the growing demand for precise, real-time location information in today's increasingly connected world [6].

The methods mentioned above face a few obstacles, such as limited accuracy, high computational complexity, and insufficient processing power, among others. Conversely, artificial intelligence (AI) and machine learning (ML) have proven successful in indoor localization due to their capacity for decision-making that does not rely on precise models.

Various ML techniques have been utilized to address non-line-of-sight (NLOS) identification and mitigation challenges. Specifically, supervised and unsupervised machine learning methods were employed in [7]–[9], while deep learning (DL) was applied for NLOS mitigation in [10]. In [11], a DL-based recursive neural network (RNN) was used to manage the fluctuations in RSSI signals by analyzing their time-domain correlations. Furthermore, DL techniques have been leveraged to uncover hidden features of RSSI measurements, helping to minimize the collection of fingerprint data in [12] and facilitate robot navigation in unknown environments in [13].

LBS technologies are vital for enabling automation within Industry 4.0. To ensure high-quality processes in this advanced industrial landscape, we propose an analysis and comparison of various supervised methods applicable to fingerprint-based systems. This examination will facilitate the selection of the most suitable approach for achieving optimal performance in diverse Industry 4.0 applications.

Fingerprint-based systems method was developed to locate devices in a specific environment using radio signals, including telecommunications parameters such as delays, signal strength, and others [14]. These systems offer advantages for location-based services (LBS), particularly in indoor environments where traditional localization technologies, such as GPS, may not be effective. One key benefit of these systems is their robustness. They can handle signal obstructions and multi-path effects that are common in indoor settings due to the presence of walls, furniture, and other objects. This robustness allows

Bismark C. Teixeira, University of Brasília, Campus Gama, Gama-DF, e-mail: bismarkcotrim@hotmail.com. Julia B. Silva, University of Brasília, Campus Gama, Gama-DF, e-mail: juliaborges.6@gmail.com. Daniel C. Araújo, University of Brasília, Campus Gama, Gama-DF, e-mail: daniel.araujo@unb.br. Diego A. Sousa, Federal Institute of Education, Science, and Technology of Ceará (IFCE), Campus Paracuru-CE, e-mail: diego.sousa@ifce.edu.br. This work was financially supported by FAPDF, under Grant Number 00193.00001046/2021-22, as part of the EDITAL 03/2021 - DEMANDA INDUZIDA. The authors gratefully acknowledge the funding and support provided by the agency.

them to maintain performance even in complex environments. These systems are also versatile, capable of utilizing various types of signals, such as Wi-Fi, Bluetooth, RFID, or even magnetic fields. This adaptability enables them to work in diverse environments and accommodate different LBS requirements.

Furthermore, once the fingerprint database has been established, these systems can efficiently manage a large number of users or devices without the need for significant additional investments in infrastructure, lowering deployment costs.

This paper proposes a solution that employs multiple access points to collect diverse measurements from a facility floor. These measurements are gathered historically from devices that have previously traversed the area. The access points connect to a central unit that processes the assembled dataset to estimate the position. This information can subsequently be utilized by robots operating within the environment.

The performance of this solution was evaluated using the QUADRIGA model, which adheres to the 3rd Generation Partnership Project (3GPP) specification. The QUADRIGA model is based on the specifications of 3GPP, which is a consortium of telecommunications associations from various countries that defines standards and recommendations in GSM and WCDMA technologies for mobile devices [15]. Additionally, our models take into account the discretization of the RSSI. By incorporating these factors, the proposed approach aims to provide a robust and accurate indoor localization system for various applications.

II. Problem Definition

This work addresses an indoor environment with I access points (APs), each with N antennas, and a single-antenna device. The locations of AP i and the user are represented as $\mathbf{p}i = [p_{x,i}, p_{y,i}]$ and $\mathbf{q} = [q_x, q_y]$, respectively. We assume that the APs are linked to a localization center, where measurements are processed to estimate the user's position.

A. Transmission Model

In this orthogonal frequency division multiplexing (OFDM) setup, after cyclic prefix removal and fast Fourier transform (FFT), the signal received at subcarrier m from the ith AP can be given as:

$$\mathbf{y}[m,i] = \sqrt{p_m}\mathbf{h}[m,i]x[m] + \mathbf{n}[m,i], \tag{1}$$

where $\mathbf{y}[m,i]$ is the received signal, p_m the power allocated to subcarrier m, and $x[m]$ a pilot symbol with $|x[m]| = 1$. $\mathbf{n}[m,i]$ is a Gaussian noise vector with zero mean and variance $\frac{N_0}{2}$. Given this model, our goal is to estimate the vector $\mathbf{q}$, representing the user's position.

The channel $\mathbf{h}[m,i]$ is a sum of various paths, with the line-of-sight path presumed as the strongest:'

$$\mathbf{h}[m,i] = \sum_{l=0}^{L-1} \rho_l e^{\jmath\phi_l}\mathbf{a}(\theta_l)e^{\jmath 2\pi m\Delta f\tau_l}. \tag{2}$$

For path loss estimation, each AP applies the OFDM signal model as follows:

$$\hat{\rho} = \left|\left|\frac{\mathbf{y}[m,i]x[m]^*}{\sqrt{p_m}}\right|\right|^2. \tag{3}$$

This information is conveyed to a central network that runs an algorithm to estimate the user's location. Traditional methods such as multilateration are often employed in this context, yet they tend to struggle with non-linear aspects, including quantization during the backhaul communication process.

Machine learning, on the other hand, emerges as a promising alternative due to its aptitude for managing non-linearity, estimation error, and multipath interference. The following section will outline the AP distribution and the system architecture within which our machine-learning algorithm operates.

III. Implementation

In this section, we delve into the core of our solution for user position estimation in indoor scenarios. Specifically, we elucidate the processes involved in constructing the dataset, detailing the methodologies involved, and how they contribute to the overall setup. Additionally, we discuss our approach to applying machine learning algorithms for position estimation, highlighting the strategic steps and logical considerations driving our techniques. By examining these components, we provide a comprehensive insight into our innovative solution, underscoring its efficacy and adaptability in facilitating reliable user position estimation in indoor environments.

A. The Dataset Construction

The construction of an accurate and comprehensive dataset forms a crucial component of our proposed solution. Serving as the primary input for the machine learning algorithm, the dataset enables the algorithm to discern patterns and make precise user position estimations. The reliability and quality of the dataset can have an impact on the system's overall performance. There are two primary methods for data acquisition in this context: site surveying and channel estimation.

The site survey method is an intensive, field-based approach where a team manually measures and records path loss at various points within the location. Concurrently, they log the corresponding user positions. These path loss measurements and their corresponding positions are then paired and recorded in a dataset, effectively mapping the indoor environment in terms of user position and path loss. While this method can provide high-quality data and detailed environmental insights, it may be time-consuming, labour-intensive, and potentially costly, particularly in large or complex environments.

Conversely, the channel estimation method employs the OFDM properties to collect user measurements, as per Eq. (1). These measurements are processed to estimate the channel and extract path loss, as described in Eq. (3). As the user moves within the environment, each position and its corresponding path loss estimate are recorded in the dataset. This approach offers cost and efficiency advantages, enabling the continuous and historical gathering of measurements during system operation. However, the data quality may be compromised due to noise and potential errors in the estimation process.

Our solution operates within a system architecture comprising multiple APs strategically positioned within the indoor environment. These APs, affixed to the ceiling and distributed throughout the area as shown in Figure 1, gather signal data

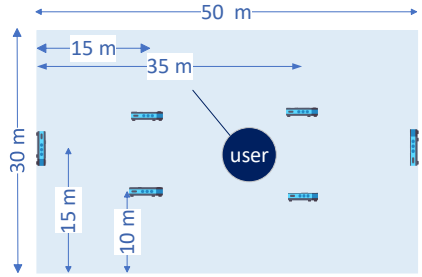


Fig. 1: Industrial warehouse with 6 APs and a single user on a straight path.

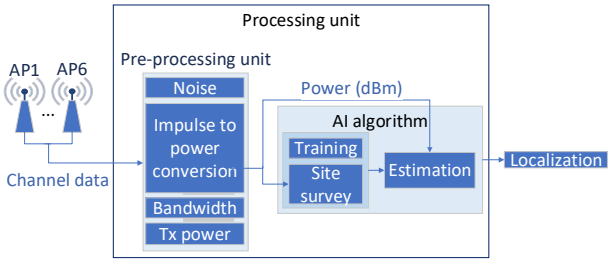


Fig. 2: Block diagram of the proposed solution.

from various points, process this information, and estimate the path loss. These estimates are then conveyed to a core network, the critical hub for data collection and processing within the system architecture.

The core network is responsible for assembling these individual estimates into a comprehensive dataset, which provides the necessary information for our machine learning algorithm to accurately estimate the user's position. The role of the core network extends to managing efficient data processing, synchronization, and storage, ensuring the dataset is both reliable and up-to-date.

The core network can be implemented either locally or in a cloud-based environment. A local setup may offer lower latency and superior data security but may require substantial resources for setup and maintenance. On the other hand, a cloud-based system has benefits such as scalability, flexibility, and simplified management. The choice between these approaches will largely hinge on factors such as data security requirements, system requirements, available resources, and operational scale.

The system architecture is purposefully designed to support the effective operation of the machine learning algorithm. The strategic positioning of the APs allows for broad coverage and data diversity, potentially enhancing user position estimation accuracy. Furthermore, the core network's role in building and maintaining the dataset ensures the machine learning algorithm is fed with high-quality, diverse, and current data to perform its task effectively. This dataset is essentially a position-path loss map, used as the foundation for the supervised machine learning algorithm.

B. User Positioning Estimate

The estimation phase, the cornerstone of our solution, leverages the power of the machine learning algorithm to predict the user's position accurately. Utilizing numerous supervised machine learning algorithms, this phase marks the culmination of the collected dataset's journey and its transition into actionable performance insight.

Supervised learning is a machine learning approach where the model is trained on a labelled dataset. In our case, the labels correspond to the known positions of the user, and the features are the path loss estimates captured by the APs. By training the model on this data, it learns to associate specific path loss estimates with precise user positions. This is akin to drawing an intricate map where path loss estimates lead to specific user positions, a mapping that the algorithm learns during the training phase.

However, path loss estimates can be prone to various disturbances, such as channel estimation errors, quantization noise, and transmission errors, especially considering that these values need to be sent over the network. These errors underline the importance of using a robust and noise-tolerant machine learning algorithm.

Specifically, the Extra Trees Regressor (ETR) is equipped to handle these disturbances. It builds multiple decision trees using the training data and amalgamates their predictions to output a final prediction. This ensemble approach enhances accuracy and provides robustness against overfitting. Considering the intricate nature of indoor positioning, marked by multipath effects and environmental complexities, the ETR is an apt candidate to be chosen among the other algorithms analyzed.

The success of the estimation phase is heavily reliant upon the quality and breadth of the dataset used for training. An exhaustive and accurate dataset culminates in a more reliable position estimation, reinforcing the significance of the dataset construction phase discussed earlier. In subsequent sections, the algorithms' performance and intricacies will be better gauged and compared.

C. The Algorithm: ETR

The ETR is an ensemble machine-learning algorithm specifically tailored for regression problems. It's a variant of the Random Forest algorithm, which functions by constructing multiple decision trees during the training phase and yields the mean prediction of the individual trees for regression problems.

An additional layer of randomness, indicated by the "Extra" in Extra Trees, sets it apart from its counterparts. Beyond the standard feature and threshold randomization used in Random Forests, the ETR introduces randomness in decision tree node splitting. Instead of calculating the optimal split point for each feature, it randomly selects a subset of potential split points and determines the best among them.

The ETR significantly reduces the model's variance through this injection of additional randomness. This added element helps prevent overfitting, ensuring the model does not become overly complex and therefore prone to errors. However, the same randomness might lead to increased bias, potentially limiting the model's ability to capture the data's underlying patterns as effectively as other algorithms that use more deterministic methods.

In terms of computational complexity, the ETR ETR generally outperforms other tree-based algorithms. The complexity of training an ETR model, which involves constructing multiple decision trees, is typical $\mathcal{O}(Nd\log(N))$ where N is the number of samples and d is the number of dimensions. However, the prediction complexity is low, only $O(\log(N))$, making it an efficient option for large datasets.

Furthermore, the ETRs ability to handle high-dimensional data gives it an edge over classical algorithms like K-Nearest Neighbors (K-NN). The K-NN algorithm, while effective in certain contexts, often struggles with high-dimensional data due to the curse of dimensionality. This phenomenon causes data points in high-dimensional space to be far apart, complicating the identification of meaningful neighbours. On the contrary, the ETR, with its randomized tree construction, can more effectively handle high-dimensional data, potentially leading to better performance.

While the ETR proves a potent solution for regression tasks, particularly those involving high-dimensional or noisy datasets, it's essential to carefully assess its performance against other algorithms for the specific problem at hand. This consideration will ensure that the chosen algorithm is the best fit for the problem's unique requirements and specifics.

In the following section, the performance of the ETR is compared to other machine learning algorithms such as AdaBoost, Bayesian Regressor, Elastic Net Regressor, and Support Vector Machines. This comparative analysis further elucidates the strengths and weaknesses of each algorithm and provides insights into their performance in the context of indoor positioning systems.

IV. Results

In evaluating our proposed solution, we considered the 28 GHz channel model with the indoor channel following the 3GPP model specification. For a realistic representation of the channel, we utilized Quadriga, a versatile 3D radio propagation simulator, to generate the channel [16].

Our analysis was based on 100 potential scenarios for the respective transmission powers, each one presenting a different setting and context for the user. In each scenario, we simulated 100 users, each one being placed and moving randomly in the environment, as depicted in Fig. 1. This offered us diverse situations to test our solution's robustness and reliability.

Our evaluation aimed to scrutinize the effectiveness of our architecture using the ETR and compare it with other algorithms commonly employed in the literature. We focused our analysis primarily on two key metrics: the Root Mean Square Error (RMSE) of the position error and the coefficient of determination, called R-squared (R2), of the position error.

The RMSE is a frequently used measure of the differences between values predicted by a model and the values observed. In this context, the RMSE for the position error can be computed as follows:

$$\text{RMSE} = \sqrt{\frac{1}{n}\sum_{i=1}^{n} ||\mathbf{q}_n - \hat{\mathbf{q}}_n||^2}, \tag{4}$$

where $\mathbf{q}_n$ represents the nth observed position of the dataset, $\hat{\mathbf{q}}_n$ is the predicted position.

The R2 metric, also known as the coefficient of determination, is a statistical measure that represents the proportion of the variance for a dependent variable that's explained by an independent variable or variables in a regression model. In this case, the R-squared of the position error can be calculated as follows:

$$R2 = 1 - \frac{\sum_{i=1}^{n} ||\mathbf{q}_n - \hat{\mathbf{q}}_n||^2}{\sum_{i=1}^{n} ||\mathbf{q}_n - \bar{\mathbf{q}}||^2}, \tag{5}$$

where $\bar{\mathbf{q}}$ is the mean position of the dataset.

Seventy percent of the data was set aside for training, and thirty percent for testing. In the following charts, the regression models estimated the distance values to compare them with the actual user positions, generating the CDF for error, mean absolute error (MAE), and R2.

Figures 3a, and 3b represent the CDF of the MAE distance of the regressors for 0, 5, 10, and 15 dBm of transmitting power, respectively. It is worth noting that Decision Tree, KNN, Extra Trees and Random Forest yield better performance, the same holding true for Figures 3c, and 3d which represent the CDF of the R2 distance of the regressor at different transmitted powers. Figure 4b shows the MAPE for the best regressors at 0 and 15 dBm. It is noted that 0 and 15 dBm yield similar results, as there is a proportional change in value between the power of the access points and the user's position in the simulated environment.

By examining these regressors through the lens of the R2 chart, specifically focusing on the optimal score region around 1, and at distinct SNR values (0 and 15 dBm respectively) as previously depicted, the comparative analysis narrows down to the four top-performing regressors: Random Forest, Decision Tree, Extra-Trees, and KNN. The Extra-Trees Algorithm emerges as the closest to achieving the ideal score. Further, the Table I delineates the MAE values at the 10%, 50%, and 90% thresholds. This comparison highlights the performance differences among the four regressors, spotlighting Extra-Trees as the primary focus.

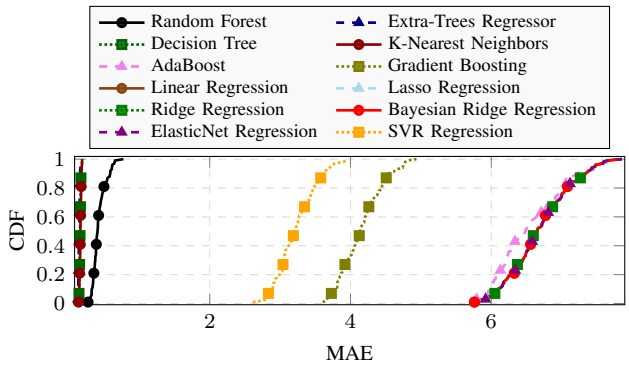
TABLE I: CDF percentile of MAE values for some of the regressors

MAE Regressors								
%	***Extra-Trees***		***K-Nearest Neigh.***		***Random Forest***		***Decision Tree***	
	0 dBm	*15 dBm*	*0 dBm*	*15 dBm*	*0 dBm*	*15 dBm*	*0 dBm*	*15 dBm*
10	0.13	0.13	0.15	0.14	0.32	0.32	0.14	0.14
50	0.14	0.14	0.16	0.16	0.40	0.40	0.16	0.16
90	0.16	0.16	0.18	0.18	0.59	0.58	0.17	0.17

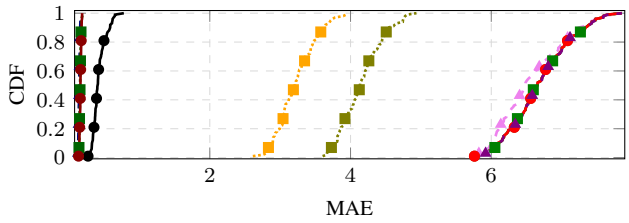
V. Conclusion

Utilizing the QUADRIGA model, we analyzed various access points in over 100 potential scenarios with differing power transmitters. This enabled a comprehensive study of indoor localization regressors and the application of the ETR. The Regressor's performance was both accurate and reliable, as validated by CDF metrics such as R2 analysis.

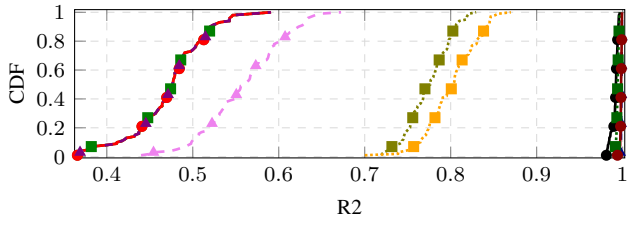
For future research, it is critical to incorporate data from actual site surveys. This will enhance our algorithm and position our regressor as an effective, real-world positioning system.



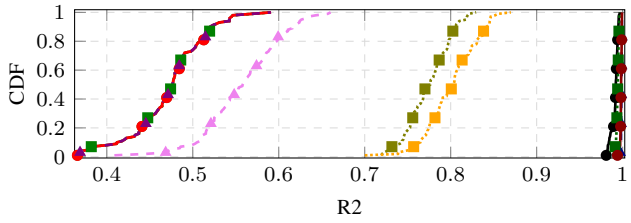
(a) Transmitting power of 0 dBm.



(b) Transmitting power of 15 dBm.



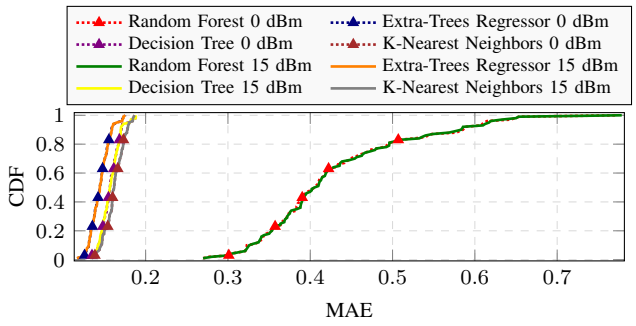
(c) Transmitting power of 0 dBm.



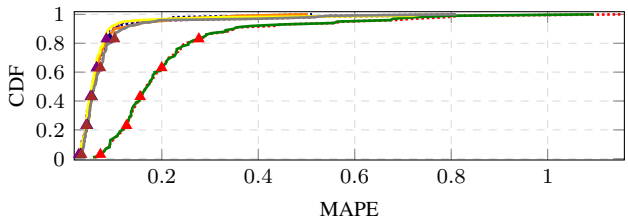
(d) Transmitting power of 15 dBm.

Fig. 3: Cumulative Distribution Function (CDF) of the Mean Absolute Error (MAE) and R-squared (R2) distance.

Furthermore, assessing and reducing the number of access points is vital for creating a realistic application simulation.



(a) Mean Absolute Error (MAE).



(b) Mean Absolute Percentage Error (MAPE).

Fig. 4: Cumulative Distribution Function (CDF) for the best four regressors from 0 and 15 dBm transmitting power.

REFERENCES

[1] A. Nessa, B. Adhikari, F. Hussain, and X. N. Fernando, "A survey of machine learning for indoor positioning," *IEEE Access*, vol. 8, pp. 214 945–214 965, 2020. DOI: 10.1109/ACCESS.2020.3039271.

[2] G. Wang, H. Chen, Y. Li, and M. Jin, "On received-signal-strength based localization with unknown transmit power and path loss exponent," *IEEE Wireless Communications Letters*, vol. 1, no. 5, pp. 536–539, 2012. DOI: 10.1109/WCL.2012.072012.120428.

[3] R. Zhang, F. Höflinger, and L. Reindl, "Tdoa-based localization using interacting multiple model estimator and ultrasonic transmitter/receiver," *IEEE Transactions on Instrumentation and Measurement*, vol. 62, no. 8, pp. 2205–2214, 2013. DOI: 10.1109/TIM.2013.2256713.

[4] Z. Yang, Z. Zhou, and Y. Liu, "From rssi to csi: Indoor localization via channel response," *ACM Comput. Surv.*, vol. 46, no. 2, Dec. 2013, ISSN: 0360-0300. DOI: 10.1145/2543581.2543592.

[5] Y. Ma, G. Zhou, and S. Wang, "Wifi sensing with channel state information: A survey," *ACM Comput. Surv.*, vol. 52, no. 3, Jun. 2019, ISSN: 0360-0300. DOI: 10.1145/3310194.

[6] S. A. (Zekavat and R. M. Buehrer, *Handbook of Position Location: Theory, Practice, and Advances, Second Edition*, 2nd. Wiley, 2018, ISBN: 9781119434610. DOI: 10.1002/9781119434610.

[7] S. Maranò, W. M. Gifford, H. Wymeersch, and M. Z. Win, "Nlos identification and mitigation for localization based on uwb experimental data," *IEEE Journal on Selected Areas in Communications*, vol. 28, no. 7, pp. 1026–1035, 2010. DOI: 10.1109/JSAC.2010.100907.

[8] M. Ramadan, V. Sark, J. Gutierrez, and E. Grass, "Nlos identification for indoor localization using random forest algorithm," in *WSA 2018; 22nd International ITG Workshop on Smart Antennas*, 2018, pp. 1–5.

[9] X. Cai, X. Li, R. Yuan, and Y. Hei, "Identification and mitigation of nlos based on channel state information for indoor wifi localization," in *2015 International Conference on Wireless Communications & Signal Processing (WCSP)*, 2015, pp. 1–5. DOI: 10.1109/WCSP.2015.7341172.

[10] C. Jiang, J. Shen, S. Chen, Y. Chen, D. Liu, and Y. Bo, "Uwb nlos/los classification using deep learning method," *IEEE Communications Letters*, vol. 24, no. 10, pp. 2226–2230, 2020. DOI: 10.1109/LCOMM.2020.2999904.

[11] M. T. Hoang, B. Yuen, X. Dong, T. Lu, R. Westendorp, and K. Reddy, "Recurrent neural networks for accurate rssi indoor localization," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 639–10 651, 2019. DOI: 10.1109/JIOT.2019.2940368.

[12] D. V. Le, N. Meratnia, and P. J. Havinga, "Unsupervised deep feature learning to reduce the collection of fingerprints for indoor localization using deep belief networks," in *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2018, pp. 1–7. DOI: 10.1109/IPIN.2018.8533790.

[13] L. Tai and M. Liu, "Mobile robots exploration through cnn-based reinforcement learning," *Robotics and Biomimetics*, vol. 3, no. 1, p. 24, 2016, ISSN: 2197-3768. DOI: 10.1186/s40638-016-0055-x.

[14] M. Nabati and S. A. Ghorashi, "A real-time fingerprint-based indoor positioning using deep learning and preceding states," *Expert Systems with Applications*, vol. 213, p. 118 889, 2023, ISSN: 0957-4174. DOI: https://doi.org/10.1016/j.eswa.2022.118889.

[15] 3GPP, *Introducing 3gpp*, 3GPP, 2023.

[16] F. Burkhardt, S. Jaeckel, E. Eberlein, and R. Prieto-Cerdeira, "Quadriga: A mimo channel model for land mobile satellite," in *The 8th European Conference on Antennas and Propagation (EuCAP 2014)*, 2014, pp. 1274–1278. DOI: 10.1109/EuCAP.2014.6902008.