Range-Angle Likelihood Maps for Indoor Positioning Using Deep Neural Networks

Muhammad Ammad[1], Paul Schwarzbach[1], Michael Schultz[2] and Oliver Michler[1]

[1]*Chair of Transport Systems Information Technology, Institute of Traffic Telematics, Technische Universität Dresden*
[2]*Chair of Air Traffic Concepts, University of the Bundeswehr Munich*

Abstract

Accurate and high precision of the indoor positioning is as important as ensuring reliable navigation in outdoor environments. Using the state-of-the-art deep learning models provides better reliability and accuracy to navigate and monitor the accurate positions in the aircraft cabin environment. We utilize the simulated aircraft cabin environment measurements and propose a residual neural network (ResNet) model to predict the accurate positions inside the cabin. The measurements include the ranges and angles between a tag and the anchors points which are then mapped onto a grid as range and angle residuals. These residual maps are then transformed into the likelihood grid maps where each cell of the grid shows the likelihood of being a true location. These grid maps along with the true positions are then passed as inputs to train the ResNet model. Since any deep learning model involve numerous parameter settings, hyperparameter optimization is performed to get the optimal parameters for training the model effectively with the highest accuracy. Once we get the best hyperparameters settings of the model, it is then trained to predict the positions which provides a centimeter-level accuracy of the localization.

Keywords

Residual Grid Maps, Range-Angle Measurements, Hyperparameter Optimization, Indoor Positioning

1. Introduction

Indoor positioning systems (IPS) have become increasingly important in recent advancements, particularly due to the limited availability of satellite positioning systems in indoor environments [1]. Technologies such as Wi-Fi, Ultra-Wideband (UWB), Bluetooth Low Energy (BLE), and ZigBee have gained attention as viable solutions for high-precision applications [2, 3]. Recently, the integration of these systems with artificial intelligence (AI) has gathered significant attention, primarily due to the increased precision and higher accuracy of their positioning outcomes [4]. Recent work has explored radio propagation simulations to improve AI-enabled localization systems by addressing conventional tool-chain bottlenecks and advancing simulation techniques [5, 6]. Building on these developments in aircraft cabin positioning systems, this study investigates simulation-based evaluation of IPS for connected aircraft cabins using state-of-the-art deep learning techniques.

As illustrated in Fig. 1, we use the data-driven localization methods utilizing the simulated range and angle of arrival (AoA) measurements based on observability from different anchor points. These measures are then used to construct the likelihood grid maps using the residuals, which serve as spatial representations of the passenger's possible location. Then the ResNet [7] model is trained on these grid maps to analyze the spatial data thus predicting the passenger's location.

The remainder of the paper is structured as follows: Section 2 reviews related work and background of different technologies. Section 3 describes the data simulation methodology, aircraft environment setup, and error modeling. Section 4 shows how range and angle residuals are calculated to generate the likelihood grid maps. Section 5 explores in-depth details of the ResNet architecture layers along with the input and loss function. Section 6 discusses the hyperparameter tuning impacts on the accuracy and finally provides a comprehensive evaluation of the model's accuracy on the dataset.

IPIN-WCAL 2025: Workshop for Computing & Advanced Localization at the Fifteenth International Conference on Indoor Positioning and Indoor Navigation, September 15–18, 2025, Tampere, Finland

Muhammad.Ammad@tu-dresden.de (M. Ammad)
0009-0007-2720-8433 (M. Ammad); 0000-0002-1091-782X (P. Schwarzbach); 0000-0003-4056-8461 (M. Schultz); 0000-0002-8599-5304 (O. Michler)

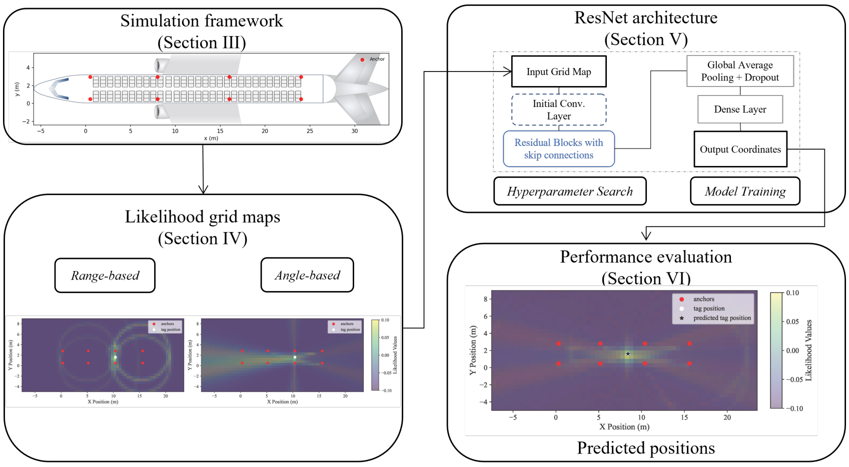


Figure 1: Graphical abstract and outline of the paper.

2. Related Work

The application of AI and machine learning (ML) technologies in indoor positioning has attracted considerable interest in recent research due to their ability to enhance accuracy, reliability and accuracy [8]. Indoor positioning methods can be categorized into two key paradigms: (1) direct AI/ML positioning, and (2) AI/ML-assisted positioning [9]. The first method estimates the location using raw sensor data (Wi-Fi, bluetooth etc.), employing ML algorithms such as neural networks, decision trees, and support vector machines to detect complex patterns without traditional signal processing [10]. This approach streamlines localization but relies on extensive training data to avoid overfitting issues which may arise during training process [4], but obtaining a robust training dataset remains a challenge.

In contrast, AI/ML-assisted positioning enhances traditional methods for localization by integrating ML for error correction and adaptive learning. This hybrid approach strengthens the reliability of conventional triangulation or fingerprint-based localization [9]. It is particularly effective for dynamic environments, where reinforcement learning and real-time adjustments enhance accuracy, optimizing wireless indoor positioning systems for reliable localization even in variable conditions [11].

The feature extraction capabilities of deep neural networks (DNNs) plays a critical role in enhancing the performance for indoor localization tasks. For instance, [12] presented a scalable DNN architecture by processing the Wi-Fi signal data to classify the multi-building and multi-floor locations, thereby reducing the localization errors typically encountered in the traditional algorithms. Furthermore, long short-term memory (LSTM) - another type of DNN - is utilized for tracking the temporal dependencies within the received signal strength (RSS) data to better meet the demands of IoT-driven localization systems [13]. Moreover, deep learning frameworks indicate that employing CNNs alongside UWB signals enhances feature extraction capabilities typically in indoor environments [14].

3. Data Simulation

The development of deep learning models for indoor positioning systems requires comprehensive training datasets that accurately reflect the complexities of real-world environments. This section details the methodology used for simulating range and angle measurements, which form the foundation for constructing the likelihood grid maps used in training the ResNet model.

In previous works the challenges for radio propagation and radio-based localization associated with the aircraft cabin environment have been studied [15, 16, 17]. To further elaborate on these challenges, we address operational simulation representing a boarding process in an A320 combined with a ray tracing and stochastical error modeling for radio propagation. The applied simulation framework for radio propagation builds upon two foundational works. Schwarzbach et al. [18] conducted an extensive survey of UWB ranging measurements in challenging environments, resulting in a statistical model. This model enables probabilistic generation of UWB range measurements with realistic characteristics, accounting for Line-of-Sight (LOS), Non-Line-of-Sight (NLOS), outlier reception scenarios, and measurement failures. The statistical distributions derived from their empirical data capture the error patterns typically encountered in real-world deployments. In addition, [6] applied this statistical model in conjunction with ray tracing to determine anchor visibility within aircraft cabin environments. This approach accounts for the complex geometry and material properties of the aircraft, providing a dataset published in [19]. Building on these works, our simulation consists of several key components: A detailed 3D model of the aircraft cabin, including geometric dimensions and material properties. Across this mode, multiple anchors are positioned throughout the cabin space based on optimal coverage analysis and practical installation constraints. Based on this, ray tracing algorithms simulate signal propagation between anchors and potential tag positions, accounting propagation phenomena introduced by cabin structures. The deterministic rays are augmented with statistical error models derived from [18], introducing realistic measurement uncertainties.

3.1. Overview of aircraft setup

The aircraft cabin environment presents unique challenges for IPS due to its elongated geometry, densely packed seating arrangement, and various obstructing elements. Our simulation recreates a standard single-aisle commercial aircraft cabin (dimensions: $30 \times 3.5 \times 2.4\,\mathrm{m}$ representing an A320. The simulation incorporates 8 anchor nodes spread throughout the plane. The configuration is visualized in Fig. 1. The reference input data for the localization system is derived from an operational simulation that models the aircraft boarding process. This simulation incorporates 148 passengers boarding the aircraft according to the methodological framework established by Schultz et al. [20]. Each passenger is assumed to carry a wireless tag, constituting the mobile nodes to be localized.

3.2. Measurement simulation and error modeling

For each passenger at each time step during the boarding process a potential tag position is derived. For this task, we use the reference positions of the boarding simulation. Based on the reference states, a true Euclidean distance d_i and a true azimuth angle θ_i between the tag positions $\mathbf{x}_i = (x_i, y_i)^\intercal$ and each anchor $\mathbf{x}^a = (x^a, y^a)^\intercal$ is calculated with:

$$d_i^a = \sqrt{(x_i - x^a)^2 + (y_i - y^a)^2} \qquad \theta_i^a = \arctan 2 \left(\frac{y_i - y^a}{x_i - x^a} \right) . \tag{1}$$

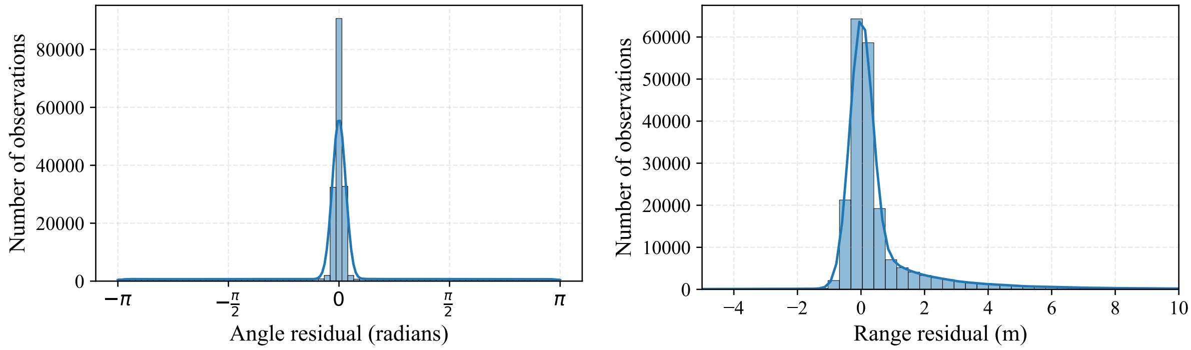
The additive error components ε_r for range measurements, are implemented based on the statistical error characterization framework established in [18]. Furthermore, the statistical properties of the AoA errors are derived from the work presented by Yu et al. in [21]. The measurement simulation follows a structured decision tree process: Beginning with the true relation (distance d, angle θ), we first determine if a measurement failure occurred. In case of failure, a null value is recorded ($r =$ None, $a =$ None). For valid measurements, the simulation classifies the error type based on the ray tracing simulation into three categories: Line-of-Sight (LOS), Non-Line-of-Sight (NLOS), or outlier conditions. Each category then applies the corresponding stochastic model from Table 1.

4. Likelihood grid maps

We implement a discrete, grid-based state space representation. The spatial domain of interest is discretized into a uniform grid G with dimensions $M \times N$, where each cell $\mathbf{g}_{y(m,n)}$ represents a position hypothesis. The fundamental advantage of using these grids come from the fully convolutional nature of the ResNet model which can innately leverage spatial representations. It can train effectively on image grids allowing the model to capture localized features and exploit spatial patterns much better than the other DNN models due to the skip connections and the residual blocks feature [22].

Table 1
Stochastic Error Models for ranging and angle measurements.

Condition	Range	Angle
LOS	$\varepsilon_r \leftarrow \mathcal{N}(0, \sigma_r^2)$ $\sigma_r = 0.3$ m	$\varepsilon_a \leftarrow \mathcal{N}(0, \sigma_\theta^2)$ $\sigma_\theta = 3°$
NLOS	$\varepsilon_r \leftarrow \mathcal{LN}(\mu, \sigma)$ $\mu = 0.8; \sigma = 1.07$	$\varepsilon_a \leftarrow \mathcal{U}(-\pi, \pi)$
Outlier	$\varepsilon_r \leftarrow \mathcal{U}(-d, d)$	$\varepsilon_a \leftarrow \mathcal{U}(-\pi, \pi)$



(a)

(b)

Figure 2: Histograms of residuals for (a) angle measurements in radians and (b) range measurements in meters.

At first, for each grid cell $\mathbf{g}_{(g,h)}$ and each anchor point $\mathbf{x}^a$, we compute the known term values based on the given geometric relationships given in Eq. (1). These values constitute the expected measurements under ideal conditions, derived directly from the geometric configuration. The residuals are then computed by comparing the known terms with the given observations ($\hat{d}$ and $\hat{\theta}$):

1. **Range residual**: $r_{(g,h)} = d_{(g,h,k)} - \hat{d}_k$
2. **Angle residual**: $r_{\theta,(g,h)} = \theta_{(g,h,k)} - \hat{\theta}_k$

These residuals quantify the discrepancy between expected and observed measurements for each potential position, forming the basis for likelihood evaluation.

The residuals are transformed into likelihood values through the application of an observation model that characterizes the probability of obtaining the observed measurement given a hypothesized position. While various models can be employed, a common approach utilizes a Gaussian model:

1. **Range likelihood**: $L^{(r)}_{(g,h,k)} = \exp\left(-\frac{(r_{(g,h,k)})^2}{2\sigma_r^2}\right)$
2. **Angle likelihood**: $L^{(\theta)}_{(g,h,k)} = \exp\left(-\frac{(r_{\theta,(g,h,k)})^2}{2\sigma_\theta^2}\right)$

where σ_r^2 and σ_θ^2 represent the variance parameters of the observation model for range and angle measurements, respectively.

The resulting likelihood maps $L^{(r)}_{(g,h,k)}$ and $L^{(\theta)}_{(g,h,k)}$ provide a spatially-resolved representation of the measurement, with higher values indicating greater likelihoods of the tag being located at the corresponding position. For anchor measurements, the individual likelihood maps can be combined through multiplication (assuming conditional independence) to obtain an overall likelihood field:

$$L_{(g,h)} = \prod_{k=1}^{K} L^{(r)}_{(g,h,k)} \cdot L^{(\theta)}_{(g,h,k)} \tag{2}$$

This integrated likelihood map serves as a comprehensive spatial representation of position probability based on the complete set of available measurements, forming the foundation for subsequent position estimation through maximum likelihood or Bayesian inference methods. An exemplary output for the derived likelihood grid maps for both measurement types is given in the graphical abstract in Fig. 1.

5. Positioning using DNNs

In this section, we explore how the deep neural networks has been implemented to estimate the location of the passengers in the simulated indoor cabin environments using the ResNet. The focus is on the model's architecture along with the inputs and loss function used for the model.

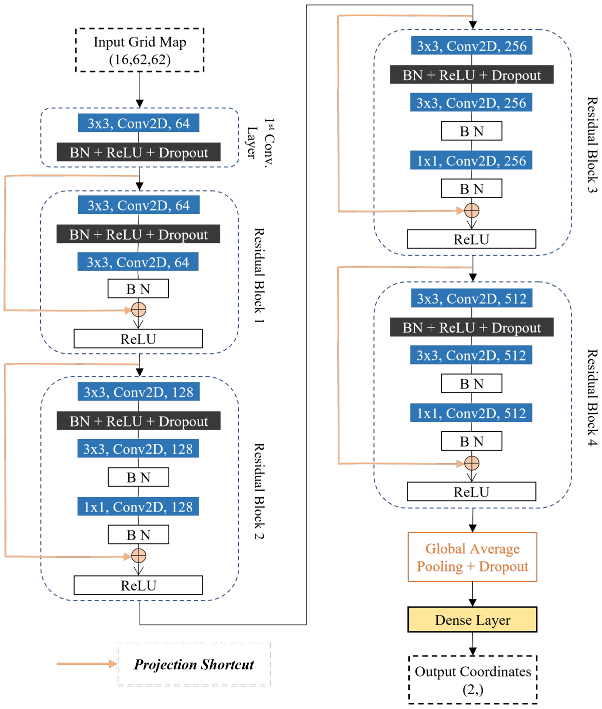


Figure 3: ResNet model architecture used for indoor localization

5.1. ResNet architecture

ResNet model has been trained on the grid maps to predict the location coordinates. The deep layering inside the ResNet model allows it to learn the complex spatial features in the likelihood grid maps. The residual blocks in the ResNet model allows the input residual maps to train more effectively with better generalization despite having a complex and deeper architecture [3]. The model's architecture shown in Fig. 3 consists of multiple residual blocks each including convolutional layers along with the batch normalization (BN) [23], rectified linear unit (ReLU) [24], and Dropout[25] layers. The addition of the dropout layers helps to mitigate overfitting enabling a better generalization for unseen data [25].

Although the original ResNet model does not include a dropout layer, we incorporate it into our architecture to further stabilize the training, as also implemented in [26]. The output of the convolutional layers inside the residual block is added back to the original input using a shortcut connection. This establishes an identity mapping [27] thus enhancing the gradient flow during back-propagation. To enable the model to predict continuous x, y position coordinates, the output layer is implemented as a Dense layer with a linear activation function.

5.2. Input and loss function

The likelihood grid maps have been used as inputs for the ResNet model as shown in Fig. 1. Each measurement have the range and AoA grid maps for all anchors but whenever there is a measurement failure, an array of zeros with the same input shape is used. This issue is commonly present in real-world scenarios due to signal blockage or measurement limitations. Handling the invalid measurement is very important to ensure consistent input dimensions for the ResNet model.

Since both grid maps share the same dimensions, these are combined by stacking them along the channel dimensions. For $\boldsymbol{N}$ anchor points, we have a total of $2 \times N$ channels. In our dataset, each measurement have ranges and angles from 8 anchors points leading to a total of 16 channels - 8 channels for each measurement's likelihood maps. Consequently, the input array has a dimension of $16 \times 16 \times 62$ as illustrated in Fig. 3, where 62×62 represents the spatial grid shape.

ResNet being a supervised algorithm, it gets the ground truth labels for training. The ground truth labels consists of the true locations represented as a (x, y) coordinates in 2D plane. Each training epoch pairs the grid maps true locations, training the model to minimize prediction error. The mean squared error (MSE) is a regression loss function which is used for our model to minimize the error. This function guides the model optimizing towards getting accurate position predictions. It calculates the average squared difference between predicted and actual coordinates [28]. It is expressed as

$$MSE = \frac{1}{n}\sum_{i=1}^{n}(y\_pred - y\_true)^2, \tag{3}$$

where *y_pred* represents the (x, y) coordinates predicted by the model and *y_true* represents the actual (x, y) coordinates.

6. Performance evaluation

To evaluate the performance of any deep learning model, it is crucial to ascertain the optimal hyperparameter settings that would enhance the model performance significantly. Hyperparameter tuning is carried out to avoid overfitting as well as optimizing the performance and generalization of machine learning models [29]. The choice of hyperparameters significantly influences the efficiency of training and accuracy of predictions. Before training the model, we implemented a systematic hyperparameter search to identify the best configurations and then train the ResNet model on those settings.

6.1. Impact of hyperparameter selection on accuracy

The hyperparameters selected include learning rate, dropout rate, batch size, and optimizer (cf. Table 2). The hyperparameter space from which these were drawn is also included, alongside the best hyperparameter settings concluded from trials based on performance outcomes shown in Table 3.

Table 2
Hyperparameter search space and selected hyperparameters for the ResNet model

Hyperparameter	Search Space	Selected Values
Optimizer	['Adam'[30], 'Adamax'[30], 'Adagrad'[31], 'RMSprop'[32]]	Adam
Learning Rate	[0.01, 0.001, 0.0005, 0.0001]	0.001
Batch Size	[8, 16, 32, 64]	32
Dropout Rate	[0.2, 0.3, 0.4, 0.5]	0.2

We have applied 12 iterations on different combinations of hyperparameters and evaluated the loss to track the model's performance based on Bayesian Optimization technique [33]. The validation loss (val_loss) (in meters) serves as an indicator of the model's capacity to generalize on unseen data.

As shown in Table 3, the model reaches its optimal performance in trial number 10 with the least val_loss, utilizing Adam as the optimizer, a learning rate of 0.001, a batch size of 32, and a dropout rate of 0.2. Using Adam an optimizer has been favorable in many deep learning applications since it adaptively adjusts the learning rate, facilitating more efficient convergence rates compared to traditional

Table 3
Hyperparameter search trials based on Bayesian Optimization with the validation loss in meters

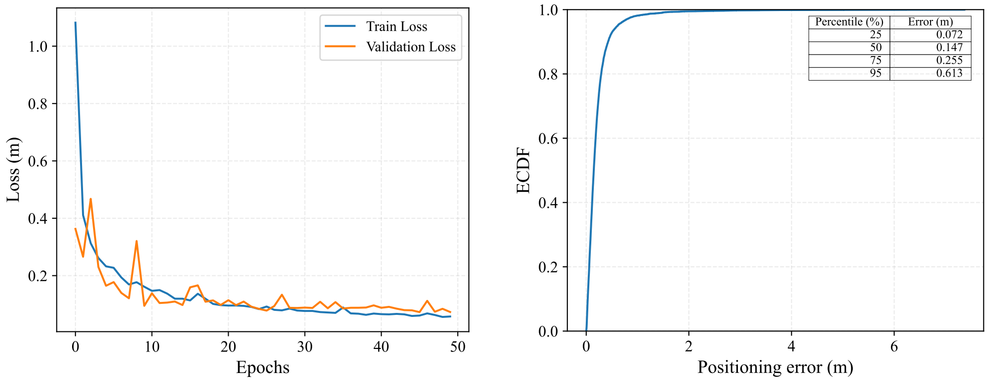
Trial No.	Optimizer	Learning Rate	Batch Size	Dropout Rate	Val_loss
1	Adagrad	0.0001	32	0.2	3.724
2	Adagrad	0.0005	8	0.3	15.149
3	Adagrad	0.0001	8	0.5	26.267
4	RMSProp	0.0005	16	0.4	0.105
5	Adamax	0.0001	64	0.3	0.448
6	RMSProp	0.01	8	0.2	0.096
7	Adam	0.0001	16	0.3	0.116
8	Adam	0.01	16	0.5	0.153
9	Adagrad	0.01	64	0.5	0.329
10	**Adam**	**0.001**	**32**	**0.2**	**0.084**
11	Adam	0.001	8	0.3	0.095
12	Adamax	0.001	16	0.3	0.106

optimizers [34]. Additionally, trial 1 to 3 show that the model performance is declined significantly with the utilization of Adagrad optimizer with smaller batch sizes.

6.2. Results

The output of our ResNet model provides the x, y coordinates of the predicted location based on the input grid maps. An example output of predicted position is shown in graphical abstract in Fig. 1. The model's performance has been assessed using test dataset comprising 25% of the total dataset, while the remaining 75% is allocated for training the ResNet model. The model is trained over 50 epochs, as illustrated in Fig. 4a, which also displays both the training and validation loss (in meters) throughout the epochs. It is observed that the loss significantly decreases as the epoch count increases; however, it begins to flatten out after approximately 30th epoch.

During the training process, we defined a custom checkpoint using Keras functionality to save the trained model if the accuracy of the subsequent epoch improves. The last saved model with the best accuracy came out to be at 46th epoch with the validation loss equal to $7.28\,\text{cm}$ on test dataset. This accuracy is consistent with, and in many cases exceeds, existing positioning accuracies for indoor localization that employs technologies such as Wi-Fi, UWB, or BLE for positioning.



Percentile (%)	Error (m)
25	0.072
50	0.147
75	0.255
95	0.613

(a) (b)

Figure 4: (a) Training and validation losses over 50 training epochs. (b) ECDF of positioning error.

Fig. 4b illustrates the positioning errors as empirical cumulative distribution function (ECDF). It shows an error of $0.147\,\text{m}$ for the 50th percentile, meaning half of the predictions fall within $14.7\,\text{cm}$. Beyond $0.613\,\text{m}$ (95th percentile), the curve plateaus, showing only a few high-error outliers.

7. Conclusion

Recent advancements in artificial intelligence have established a new potential in indoor localization by providing a reliable and accurate positioning in indoor environments. In this study, we train ResNet model to learn the underlying spatial patterns in the likelihood grid maps generated based on the simulated range and AoA measurements. The likelihood grid maps, which are used as model inputs, represent the pixel-level position hypothesis. Before training the model, hyperparameter tuning is performed to get the best training parameters since DNNs always involve multiple parameter settings which can tweak the positioning performance. The attainment of a model's accuracy of $7.28\,\mathrm{cm}$ signifies a potential for advanced positioning in indoor environments. The trained model is then used to predict positions on test dataset and the results showed that 95% of the predictions have a positioning error less than $0.613\,\mathrm{m}$, with a median error of only $0.147\,\mathrm{m}$. This level of precision underscores the effectiveness of our approach and lays a robust groundwork for future enhancement in indoor localization systems.

Acknowledgments

This work has been funded by the German Federal Ministry for Economic Affairs and Climate Action (BMWK) within the project INTACT (FKZ: 20D2128D).

Gefördert durch:

Bundesministerium
für Wirtschaft
und Klimaschutz

aufgrund eines Beschlusses
des Deutschen Bundestages

Declaration on Generative AI

The author(s) have not employed any Generative AI tools.

References

[1] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, E. Aboutanios, Recent advances in indoor localization: A survey on theoretical approaches and applications, IEEE Communications Surveys & Tutorials 19 (2017) 1327–1346. doi:10.1109/comst.2016.2632427.

[2] C.-H. Hsieh, J.-Y. Chen, B.-H. Nien, Deep learning-based indoor localization using received signal strength and channel state information, IEEE Access 7 (2019) 33256–33267. doi:10.1109/access.2019.2903487.

[3] Z. Niu, H. Yang, L. Zhou, M. Farag Taha, Y. He, Z. Qiu, Deep learning-based ranging error mitigation method for uwb localization system in greenhouse, Computers and Electronics in Agriculture 205 (2023) 107573. doi:10.1016/j.compag.2022.107573.

[4] A. Nessa, B. Adhikari, F. Hussain, X. N. Fernando, A survey of machine learning for indoor positioning, IEEE Access 8 (2020) 214945–214965. doi:10.1109/access.2020.3039271.

[5] A. Michler, J. Ninnemann, J. Krauthäuser, P. Schwarzbach, O. Michler, Potentials of deterministic radio propagation simulation for ai-enabled localization and sensing, in: 2023 13th International Conference on Indoor Positioning and Indoor Navigation (IPIN), IEEE, 2023. doi:10.1109/ipin57070.2023.10332482.

[6] P. Schwarzbach, J. Ninnemann, H. Ußler, O. Michler, M. Schultz, Simulation-based evaluation of indoor positioning systems in connected aircraft cabins, Transportmetrica B: Transport Dynamics 12 (2024). doi:10.1080/21680566.2024.2321454.

[7] K. He, X. Zhang, S. Ren, J. Sun, Deep residual learning for image recognition, 2015. doi:10.48550/ARXIV.1512.03385.

[8] S. Pešić, Modern Challenges in Indoor Positioning Systems: AI to the Rescue, IntechOpen, 2024. doi:10.5772/intechopen.1005354.

[9] Y. Yang, M. Chen, Y. Blankenship, J. Lee, Z. Ghassemlooy, J. Cheng, S. Mao, Positioning using wireless networks: Applications, recent progress and future challenges, 2024. arXiv:2403.11417.

[10] S. Bozkurt, G. Elibol, S. Gunal, U. Yayan, A comparative study on machine learning algorithms for indoor positioning, in: 2015 International Symposium on Innovations in Intelligent SysTems and Applications (INISTA), IEEE, 2015, p. 1–8. doi:10.1109/inista.2015.7276725.

[11] J. L. Carrera Villacres, Z. Zhao, T. Braun, Z. Li, A particle filter-based reinforcement learning approach for reliable wireless indoor positioning, IEEE Journal on Selected Areas in Communications 37 (2019) 2457–2473. doi:10.1109/jsac.2019.2933886.

[12] K. S. Kim, S. Lee, K. Huang, A scalable deep neural network architecture for multi-building and multi-floor indoor localization based on wi-fi fingerprinting, Big Data Analytics 3 (2018). doi:10.1186/s41044-018-0031-2.

[13] Q. Ye, X. Fan, G. Fang, H. Bie, Exploiting temporal dependency of rss data with deep learning for iot-oriented wireless indoor localization, Internet Technology Letters 6 (2022). doi:10.1002/itl2.366.

[14] L. Nosrati, S. H. Semnani, M. S. Fazel, S. Rakhshani, M. Ghavami, Improving indoor localization using mobile uwb sensor and deep reinforcement learning, IEEE Sensors Journal (2024). doi:10.1109/jsen.2024.3442974.

[15] P. Schwarzbach, J. Engelbrecht, A. Michler, M. Schultz, O. Michler, Evaluation of technology-supported distance measuring ensure safe aircraft boarding during covid-19 pandemic, Sustainability 12 (2020). doi:10.3390/su12208724.

[16] F. Geyer, D. Schupke, Precise onboard aircraft cabin localization using UWB and ML, in: IEEE GLOBECOM, 2022, pp. 2032–2037. doi:10.1109/GLOBECOM48099.2022.10000651.

[17] J. F. Schmidt, D. Neuhold, C. Bettstetter, J. Klaue, D. Schupke, Wireless connectivity in airplanes, IEEE Access 9 (2021) 52913–52925. doi:10.1109/ACCESS.2021.3070141.

[18] P. Schwarzbach, R. Weber, O. Michler, Statistical evaluation and synthetic generation of ultra-wideband distance measurements for indoor positioning systems, IEEE Sensors Journal 22 (2022). doi:10.1109/jsen.2021.3121627.

[19] P. Schwarzbach, J. Ninnemann, Simulation data: Indoor positioning systems in connected aircraft cabins, 2023. doi:10.25532/OPARA-275.

[20] M. Schultz, Fast Aircraft Turnaround Enabled by Reliable Passenger Boarding, Aerospace 5 (2018) 8.

[21] K. Yu, Y. Guo, Statistical nlos identification based on aoa, toa, and signal strength, IEEE Transactions on Vehicular Technology 58 (2009) 274–286. doi:10.1109/tvt.2008.924975.

[22] A. Conneau, H. Schwenk, L. Barrault, Y. Lecun, Very deep convolutional networks for text classification, in: Proceedings of the 15th Conference of the European Chapter of the Association for Computational Linguistic, 2017. doi:10.18653/v1/e17-1104.

[23] S. Ioffe, C. Szegedy, Batch normalization: Accelerating deep network training by reducing internal covariate shift, 2015. doi:10.48550/ARXIV.1502.03167.

[24] A. F. Agarap, Deep learning using rectified linear units (relu), 2018. doi:10.48550/ARXIV.1803.08375.

[25] N. Srivastava, G. Hinton, A. Krizhevsky, I. Sutskever, R. Salakhutdinov, Dropout: A simple way to prevent neural networks from overfitting, Journal of Machine Learning Research 15 (2014) 1929–1958. URL: http://jmlr.org/papers/v15/srivastava14a.html.

[26] B. J. Kim, H. Choi, H. Jang, D. Lee, S. W. Kim, How to use dropout correctly on residual networks with batch normalization, 2023. doi:10.48550/ARXIV.2302.06112.

[27] K. He, X. Zhang, S. Ren, J. Sun, Identity mappings in deep residual networks, 2016. doi:10.48550/ARXIV.1603.05027.

[28] J. Terven, D. M. Cordova-Esparza, A. Ramirez-Pedraza, E. A. Chavez-Urbiola, J. A. Romero-Gonzalez, Loss functions and metrics in deep learning, 2023. doi:10.48550/ARXIV.2307.02694.

[29] J. A Ilemobayo, O. Durodola, O. Alade, O. J Awotunde, A. T Olanrewaju, O. Falana, A. Ogungbire, A. Osinuga, D. Ogunbiyi, A. Ifeanyi, I. E Odezuligbo, O. E Edu, Hyperparameter tuning in machine learning: A comprehensive review, Journal of Engineering Research and Reports 26 (2024) 388–395. doi:10.9734/jerr/2024/v26i61188.

[30] D. P. Kingma, J. Ba, Adam: A method for stochastic optimization, 2014. doi:10.48550/ARXIV.1412.6980.

[31] J. Duchi, E. Hazan, Y. Singer, Adaptive subgradient methods for online learning and stochastic optimization, Journal of Machine Learning Research 12 (2011) 2121–2159.

[32] T. Tieleman, G. Hinton, Lecture 6.5 - rmsprop: Divide the gradient by a running average of its recent magnitude, 2012. COURSERA: Neural Networks for Machine Learning.

[33] G. Onorato, Bayesian optimization for hyperparameters tuning in neural networks, 2024. doi:10.48550/ARXIV.2410.21886.

[34] P. Zhou, X. Xie, Z. Lin, S. Yan, Towards understanding convergence and generalization of adamw, IEEE Transactions on Pattern Analysis and Machine Intelligence 46 (2024) 6486–6493. doi:10.1109/tpami.2024.3382294.