

Optimal Hyperparameters Solutions for DNN to Improve Accuracy of Indoor Positioning Systems Based on Channel Impulse Response

Thi-Kieu-Lan Ta¹, Van-Phuc Hoang^{1†}, Van-Lan Dao², and Thi-Yen Hoang¹, Non-members

ABSTRACT

This paper introduces a fault-tolerant indoor robot localization framework that leverages deep neural networks (DNNs) to process channel impulse response (CIR) data from multiple access points (APs). Unlike conventional CIR-based localization methods that assume stable infrastructure, this work explicitly addresses the challenge of AP failure—an issue largely overlooked in prior studies. Two complementary DNN architectures are investigated: a single-input model that concatenates CIRs from all APs, and a modular multi-input model in which each branch processes the CIR of an individual AP. To improve robustness, we propose a hybrid Hyperband–Simulated Annealing (SA) optimization strategy for both hyperparameters and network parameters. Experiments show that while the single-input network achieves high accuracy under ideal conditions (ADE = 0.552 m), its performance deteriorates significantly under AP outages (up to 1.019 m). In contrast, the multi-input architecture—with branch-specific hyperparameter tuning—consistently maintains strong performance across all failure scenarios, achieving ADEs of 1.15–1.19 m. These results demonstrate the novelty and effectiveness of combining architectural modularity with adaptive optimization to achieve resilient CIR-based localization in dynamic indoor environments.

Keywords: Indoor Positioning, Convolutional Neural Network, Channel Impulse Response, Simulated Annealing, Hyperband, Hyperparameter Optimization

1. INTRODUCTION

Indoor localization enables critical applications in autonomous robotics, smart environments, and asset tracking [1], [2]. Unlike GPS, indoor systems face challenges from multipath propagation, signal attenuation, and non-line-of-sight conditions. Channel Impulse

Response (CIR) provides rich spatial information for accurate positioning, while Deep Neural Networks (DNNs) have significantly enhanced CIR-based localization performance. However, real-world deployment presents challenges, particularly access point (AP) failures that can substantially degrade accuracy [3].

Traditional approaches include Received Signal Strength Indicator (RSSI)-based methods [4, 5], which are simple but suffer from low accuracy and multipath vulnerability [6], and Channel State Information (CSI)-based methods [7], which improve precision at higher hardware complexity. Techniques span signal-, time-, and angle-based categories, with RSSI fingerprinting widely adopted for simplicity [8, 9], while Time of Arrival (ToA)/Time Difference of Arrival (TDoA) require synchronization [10] and Angle of Arrival (AoA) requires directional antennas [11].

Machine learning advances include K-Nearest Neighbors (KNN) and Support Vector Machines (SVMs). However, these models struggle in complex scenarios [12, 13]. Deep learning architectures such as Convolutional Neural Networks (CNNs) and Recurrent Neural Networks (RNNs) have shown strong capability in handling noisy inputs and extracting spatial-temporal features [14, 15]. Hybrid models achieve sub-meter accuracy [16], while graph-based and transfer-learning approaches improve cross-environment robustness [17, 18]. Existing research predominantly assumes stable infrastructure configurations, overlooking the practical reality of AP failures.

This study examines two DNN architectures designed to preserve localization accuracy during AP outages through strategic hyperparameter and parameter optimization. We assess fault tolerance under individual AP failure conditions using: (1) a single-input DNN architecture employing Hyperband-optimized hyperparameters tailored to specific failure scenarios, and (2) a multi-input DNN framework featuring dedicated processing branches for each AP, enabling adaptive exclusion of compromised channels. The experimental platform includes a mobile robot operating within an indoor environment equipped with four APs (Fig. 1). Channel modeling incorporates ray-tracing simulation to capture multipath propagation [19]. Three multi-input configurations are evaluated: uniform hyperparameters for all branches, globally optimized hyperparameters, and branch-specific optimization. Hyperparameter and parameter search employs Hyperband and Simulated Annealing (SA), with Average Distance Error (ADE) as

Manuscript received on December 7, 2025; revised on February 4, 2026; accepted on March 10, 2026. This paper was recommended by Associate Editor Kampol Woradit.

¹The authors are with Le Quy Don Technical University, Hanoi, Vietnam.

²The author is with Hitachi Energy AB, Sweden.

[†]Corresponding author: phuchv@lqdtu.edu.vn

©2026 Author(s). This work is licensed under a Creative Commons Attribution-NonCommercial-NoDerivs 4.0 License. To view a copy of this license visit: <https://creativecommons.org/licenses/by-nc-nd/4.0/>.

Digital Object Identifier: 10.37936/ecti-ec.2026242.262701

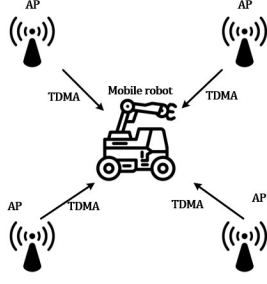


Fig. 1: Communication Model Between The Mobile Robot And APs Using Time Division Multiple Access (TDMA).

the primary metric. Results show that branch-specific optimization yields the highest robustness under AP failures.

The remainder of this paper is organized as follows. Section 2 presents the proposed method, beginning with the single-input DNN architecture followed by the multi-input architecture and its hybrid optimization strategy. Section 3 describes the experimental setup. Section 4 discusses the results, including: (i) performance of the single-input DNN under AP failures, (ii) comparison among multi-input configurations, and (iii) comparison between single-input and multi-input DNNs. Section 5 concludes the study.

2. PROPOSED METHOD

Let $\mathbf{C}_i \in \mathbb{R}^T$ denote the CIR vector from the i -th AP. At each timestep, the input is a set of CIRs from the four APs, as shown in (1).

$$\mathbf{X} = \{\mathbf{C}_1, \mathbf{C}_2, \mathbf{C}_3, \mathbf{C}_4\} \quad (1)$$

The objective is to learn the function $f(\mathbf{X})$ that maps the input $\mathbf{X}$ to the robot's 2D position:

$$f(\mathbf{X}) \approx (\mathbf{x}, \mathbf{y}) \quad (2)$$

In case of AP failure (e.g., missing $\mathbf{C}_i$), the system input becomes:

$$\mathbf{X}' = \{\mathbf{C}_1, \dots, \emptyset, \dots, \mathbf{C}_4\} \quad (3)$$

and the localization model must still maintain accurate performance.

The ADE metric is used to evaluate localization accuracy, defined as:

$$\text{ADE} = \frac{1}{M_{\text{test}}} \sum_{j=1}^{M_{\text{test}}} \sqrt{(x_{m,j} - \hat{x}_{m,j})^2 + (y_{m,j} - \hat{y}_{m,j})^2} \quad (4)$$

where $(x_{m,j}, y_{m,j})$ is the ground truth position and $(\hat{x}_{m,j}, \hat{y}_{m,j})$ is the estimated position.

Two deep learning-based approaches are implemented for indoor robot localization using CIR inputs from four APs, with built-in robustness against AP failures. The first approach employs a single-input DNN

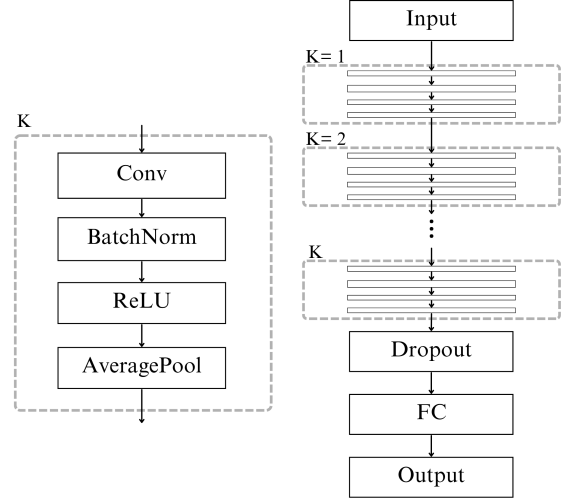


Fig. 2: Single-input DNN architecture.

architecture, presented in Subsection 2.1 The second approach adopts a modular multi-input DNN architecture that dynamically adapts to varying AP availability during inference, as described in Subsection 2.2

2.1 Single-Input DNN Architecture

The first approach employs a conventional feedforward DNN that processes concatenated CIR vectors from all four APs. Specifically, each CIR sequence $\mathbf{C}_i \in \mathbb{R}^T$ is stacked to form a single input vector $\mathbf{x} \in \mathbb{R}^{4T}$, where $\mathbf{x} = [\mathbf{C}_1^T, \mathbf{C}_2^T, \mathbf{C}_3^T, \mathbf{C}_4^T]^T$. The architecture, illustrated in Fig. 2, comprises seven hidden layers followed by a regression layer that outputs the estimated 2D location $(\hat{x}, \hat{y})$.

To ensure robustness under varying AP fault scenarios, the Hyperband algorithm is utilized to optimize the network's key hyperparameters, including the number of convolutional filters (ranging from 32 to 512), learning rates (from 0.1 to 0.5), and dropout rates (between 10^{-4} and 10^{-2}). Each hyperparameter configuration is trained for a total of 30 epochs, with intermediate evaluations every 10 epochs to guide the Hyperband selection process.

The hyperparameter tuning is performed independently for five distinct scenarios: one with no AP failure, and four cases in which each of the APs fails individually. In failure scenarios, the CIR vector of the failed AP is replaced with a zero-padded vector, and a dedicated model trained under the same fault condition is deployed during inference.

2.2 Multi-Input DNN Architecture

The second approach employs a modular multi-input DNN architecture, in which each AP is assigned a dedicated processing branch. Each branch mirrors the structure of the single-input DNN and extracts local features from its corresponding CIR input. The outputs of all active branches are concatenated and passed through a shared fully connected subnetwork to estimate the

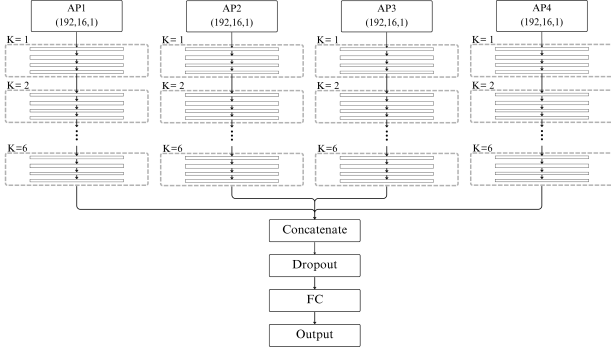


Fig. 3: Multi-input DNN architecture.

2D position, as shown in Fig. 3. This design enables dynamic exclusion of branches corresponding to faulty APs without requiring model retraining.

Three configurations are investigated for this architecture. In Configuration 1 (Fixed Parameters), all branches share the same network structure and a single fixed set of hyperparameters (e.g., units = (256,...,256), dropout = 0.4, learning rate = 0.001), regardless of which AP is active or faulty. This configuration does not adapt to AP failures and is used as the baseline for comparison.

In Configuration 2 (Fault-Aware Shared Parameters), all active branches still share identical hyperparameters; however, this shared set varies depending on the AP failure scenario. Specifically, for each failure case, the branches adopt the Hyperband-optimized hyperparameters obtained from the corresponding single-input model. For example, if AP2 is unavailable, every remaining branch uses the hyperparameters optimized for the "AP2-failure" scenario.

In Configuration 3 (Independently Tuned Parameters), each branch is assigned its own set of hyperparameters, independently optimized using Hyperband. As a result, the branches no longer share a common configuration; instead, each adapts to the characteristics and reliability of its own CIR input. This configuration offers the highest flexibility and provides enhanced robustness under heterogeneous or fault-prone conditions.

2.3 Optimization Strategies

To enhance the performance and robustness of the proposed DNN-based indoor localization system, two complementary optimization techniques are employed at different levels of the framework, namely Hyperband and SA, as illustrated in Fig. 4.

Hyperband is utilized to optimize the DNN architecture and hyperparameters for both single-input and multi-input network configurations under various AP failure scenarios. Specifically, Hyperband systematically explores the hyperparameter space by iteratively training and evaluating candidate DNN configurations on subsets of the training data. At each iteration, the ADE is computed on the validation set, and the best-performing configurations are retained. This process continues until

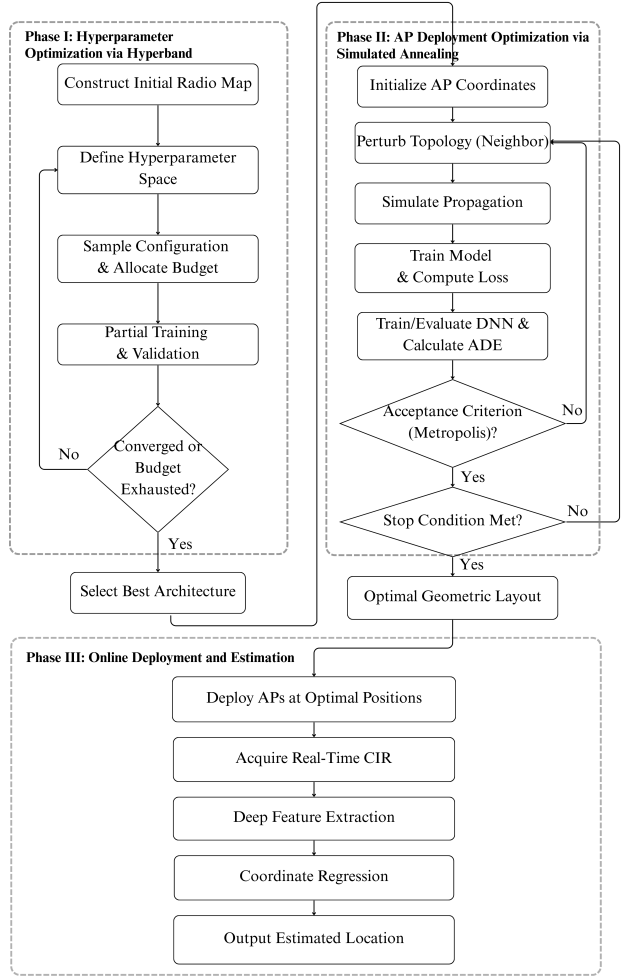


Fig. 4: Flowchart of the proposed framework integrating Hyperband for architecture search and Simulated Annealing for AP placement, with iterative ray-tracing and DNN retraining.

convergence, resulting in an optimized DNN architecture and hyperparameter setting for each considered AP fault condition.

In the second stage, SA is employed to optimize the spatial placement of APs. As depicted in Fig. 4, for each candidate AP configuration generated by the algorithm, the radio propagation environment is re-simulated via ray-tracing to produce a new channel dataset. The DNN architecture obtained from the Hyperband stage is then trained using the Adam optimizer, and the resulting ADE serves as the objective function. The SA acceptance criterion governs the selection process, facilitating effective exploration of the search space to prevent premature convergence.

3. EXPERIMENTAL SETUP

We simulate the indoor wireless channel using a ray tracing model in *MATLAB*, accounting for reflections, diffractions, and scattering via Fresnel equations and Uniform Theory of Diffraction (UTD). This yields realistic CIRs for localization. The indoor layout follows [20],

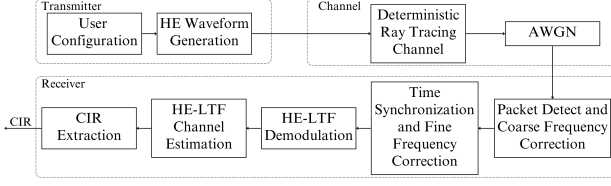


Fig. 5: CIR acquisition process.

with a 5×3.5 m² region divided into four sub-regions, each with one AP. CIR acquisition between APs and Stations (STAs) is shown in Fig. 5. IEEE 802.11az [21] is adopted, with packet transmission over AWGN channels. TDMA avoids inter-AP interference. A CIR becomes zero when Signal-to-Noise Ratio (SNR) drops below a threshold. We simulate SNR levels of 20, 25, and 30 dB. We use a bandwidth of 160 MHz and robot velocity $(v_x, v_y) = (1, 1)$ m/s. APs are fixed at 2.9 m height with known coordinates. A total of 4,851 samples were generated for the experiment. The dataset was randomly partitioned into 80% training, 10% validation, and 10% testing subsets. In addition to localization accuracy, run-time performance was evaluated by measuring inference latency using a standardized timing protocol. For each model, a single test sample corresponding to one location was first forwarded through the network for warm-up. The predict function was then repeatedly executed 100 times on the same sample, and the average per-sample inference time was reported. This procedure measures steady-state inference latency and excludes data loading or preprocessing overhead. Simulations run on a Windows 11 with Intel Core i7-14700 (28 threads, 2.1 GHz), 16 GB RAM, and NVIDIA RTX 3050 OEM GPU.

4. RESULTS AND DISCUSSION

This section evaluates the performance of the proposed localization models under various conditions of AP availability and network configurations. The analysis is organized around four main perspectives to assess both accuracy and robustness.

We begin by assessing the baseline performance of the single-input DNN when all four APs are operational, and compare it to cases where one AP is non-functional. This highlights the model's sensitivity to missing input data. Subsequently, we analyze the effect of each individual AP failure on localization accuracy, revealing their varying degrees of importance in maintaining performance. The second part focuses on the multi-input DNN architecture, emphasizing its resilience in the presence of AP faults. Finally, we compare the overall performance between single-input and multi-input networks to underscore the benefits and trade-offs associated with each design.

4.1 Single-Input DNN Optimized for Faulty AP Cases

To improve fault tolerance, separate single-input DNN models are trained for each AP failure scenario using the Hyperband algorithm. As reported in Table 1, the model

Table 1: Optimized hyperparameters and ADE at the 200th iteration of the 7-block model when one AP fails.

Faulty AP	Hyperparameter	Network Parameters	ADE at 200 Iterations
4 APs fine	units: (64, 128, 32, 256, 64, 32, 128) dropout: 0.1 learning_rate: 0.0001	393,410	0.552
AP1	units: (128, 64, 32, 32, 128, 512, 128) dropout: 0.1 learning_rate: 0.0001	1,328,130	0.776
AP2	units: (32, 256, 256, 512, 512, 512, 128) dropout: 0.3 learning_rate: 0.0001	7,164,322	0.842
AP3	units: (32, 256, 256, 64, 64, 256, 64) dropout: 0.3 learning_rate: 0.0001	2,614,498	1.019
AP4	units: (128, 512, 128, 32, 256, 32, 256) dropout: 0.2 learning_rate: 0.0001	1,450,050	0.858

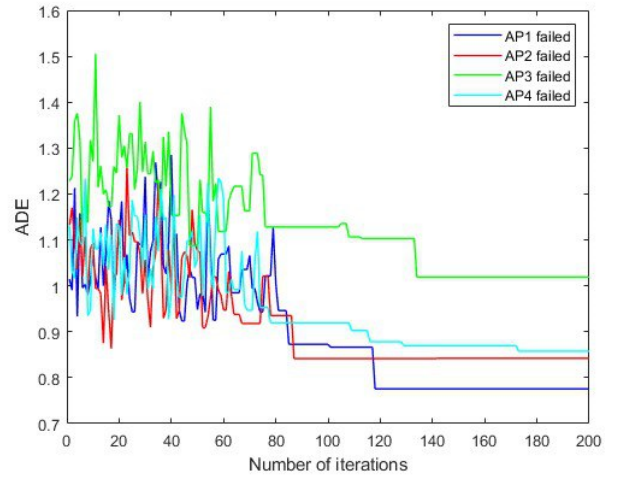


Fig. 6: Convergence of single-input DNN models under individual AP failure scenarios.

achieves an ADE of 0.552 m under normal operation with all APs functioning. However, when a single AP fails, ADE rises to 0.776, 0.842, 1.019, and 0.858 m for AP1 to AP4, respectively, with AP3 failure resulting in the highest error. This indicates AP3's critical contribution to spatial resolution.

Figure 6 depicts the training convergence patterns, showing that all models achieve stability within 100 iterations, although the AP3-failure scenario demonstrates the greatest convergence instability. The computational complexity varies considerably across failure scenarios—the AP2-failure model demands over 7 million parameters while the AP1-failure model requires only 1.3 million, highlighting the disparate spatial significance of individual APs. Even with increased architectural complexity, none of the failure-condition models match the baseline performance achieved under normal operation. This performance degradation correlates with AP positioning: AP1 and AP4, located in proximity to corners and metallic infrastructure, experience heightened multipath interference, resulting in amplified localization errors when these nodes are eliminated.

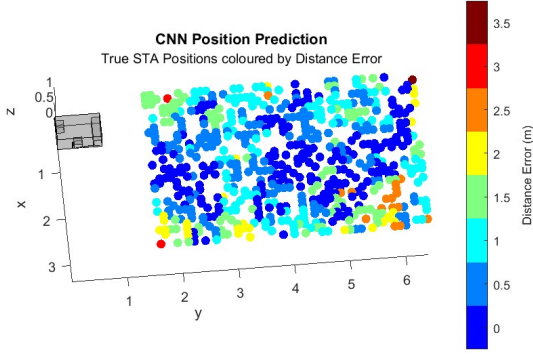


Fig. 7: Localization errors under AP1 failure with optimized AP placement (Single-input DNN).

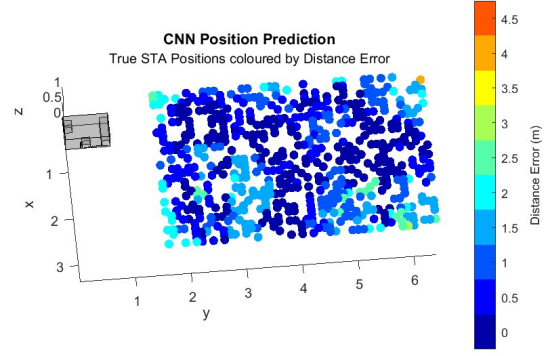


Fig. 8: Localization errors under AP2 failure with optimized AP placement (Single-input DNN).

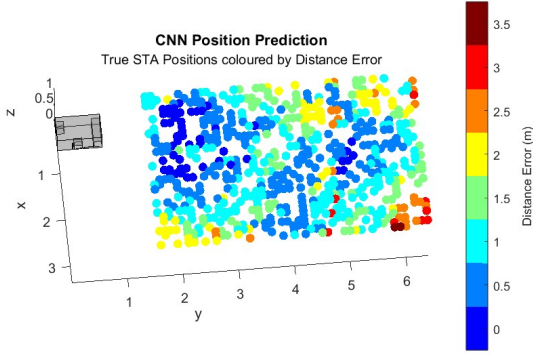


Fig. 9: Localization errors under AP3 failure with optimized AP placement (Single-input DNN).

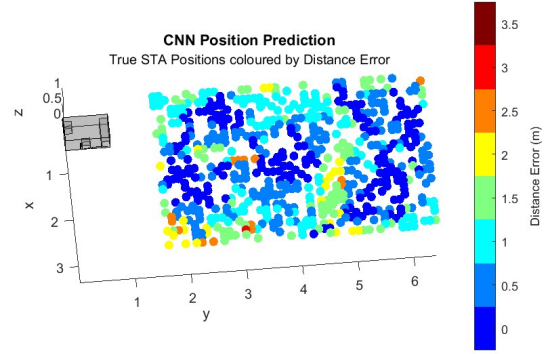


Fig. 10: Localization errors under AP4 failure with optimized AP placement (Single-input DNN).

The four-AP configuration employed in this study was optimized for maximum coverage in our prior research [20]. Spatial error distributions for individual AP failure scenarios are presented in Figures 7–10. Although most operational areas maintain acceptable accuracy levels, significant localization errors reaching 3.5 m are concentrated at room boundaries and corner regions, particularly pronounced during AP1 and AP3 outages. This behavior is primarily attributed to unfavorable geometric and propagation conditions in these marginal areas, where the remaining active APs are often located predominantly on one side of the target. Such asymmetric AP geometry significantly reduces spatial diversity, leading to ambiguous CIR signatures compounded by strong wall-induced multipath reflections and shadowing effects. These findings reveal the inherent vulnerability of single-input DNN architectures to partial AP unavailability, even under optimal geometric deployment. The observed performance degradation confirms that while single-input DNNs excel under nominal operating conditions, their localization fidelity deteriorates markedly in edge and corner regions when AP redundancies are compromised. This necessitates

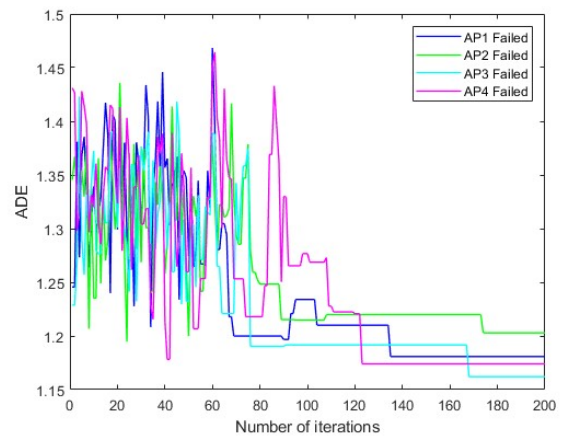


Fig. 11: Convergence of multi-input DNN models under individual AP failure scenarios using fixed hyperparameters for all branches.

investigating more resilient architectures, specifically the multi-input DNN approach examined in the next subsection.

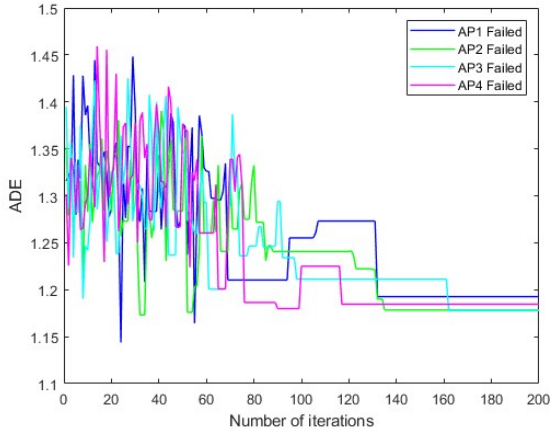


Fig. 12: Convergence of multi-input DNN models under individual AP failure scenarios with jointly optimized hyperparameters.

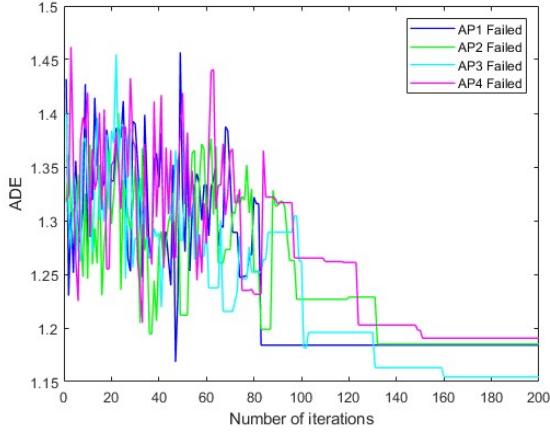


Fig. 13: Convergence of multi-input DNN models under individual AP failure scenarios using independently optimized hyperparameters for each branch.

4.2 Multi-Input DNN Optimized for Faulty AP Cases

This subsection evaluates the performance of a multi-input DNN architecture under individual AP failure scenarios, considering three configurations with progressively increasing flexibility in hyperparameter adaptation. In the first configuration, all input branches are identically configured using fixed hyperparameters (256 convolutional filters, dropout rate of 0.2, and learning rate of 0.0001). As illustrated in Fig. 11, the training process demonstrates slow convergence and notable instability across all failure cases. Final ADE values range from 1.16 m to 1.21 m, indicating limited adaptability due to the uniform treatment of heterogeneous input conditions.

The second configuration improves robustness by employing jointly optimized hyperparameters for each failure scenario, using values derived from prior single-input experiments (Table 1). Fig. 12 shows that this

Table 2: Optimized hyperparameters of multi-input DNN when one AP fails.

Faulty AP	Number of Filters	Drop-out	Learning Rate	Network Parameters
AP1	First: (512, 256, 512, 256, 256, 32, 128) Second: (128, 32, 32, 256, 256, 128, 128) Third: (128, 64, 256, 64, 256, 128, 64)	0.2	0.001	6,943,626
AP2	First: (256, 64, 32, 64, 64, 32, 128) Second: (32, 32, 128, 128, 256, 512, 32) Third: (32, 512, 512, 128, 128, 512, 256)	0.2	0.001	7,960,938
AP3	First: (128, 64, 512, 32, 32, 64, 64) Second: (64, 512, 32, 128, 256, 128, 64) Third: (512, 32, 512, 64, 64, 256, 64)	0.2	0.0001	3,055,082
AP4	First: (512, 64, 32, 128, 256, 512, 128) Second: (128, 128, 512, 512, 128, 512, 128) Third: (32, 128, 128, 256, 32, 256, 256)	0.5	0.001	9,550,506

approach enhances training stability and reduces the ADE to the range of 1.17 m–1.19 m, with convergence typically achieved before 160 iterations. Nevertheless, the spatial error distributions indicate that residual localization errors persist near room boundaries and corner regions. This suggests that while sharing parameters across branches improves overall robustness, it still restricts the model’s capacity to fully adapt to the spatial irregularities and non-linear signal distortions introduced by asymmetric AP availability.

The third configuration provides the most granular adaptation by independently tuning hyperparameters for each input branch, based on the specific AP failure condition, as detailed in Table 2. As shown in Fig. 13, this design yields the most stable training dynamics and the lowest ADEs, ranging from 1.15 m to 1.19 m. Convergence is achieved more rapidly, with reduced fluctuations, particularly within the first 130 iterations. These results confirm that branch-level optimization enables the network to better accommodate spatial and signal quality disparities caused by AP loss.

A comparative summary reveals that increased hyperparameter flexibility correlates with improved performance. The fixed-parameter model exhibits the poorest localization accuracy and convergence behavior. Shared optimization offers moderate improvements, while individualized tuning at the branch level results in the most robust and precise predictions. The corresponding spatial error maps for the third configuration under different AP failure conditions are shown in Figures 14–17. In all cases, prediction errors slightly increase along map edges. The AP4 failure case demonstrates relatively higher errors, including some near the center. However, severe errors (yellow/red regions) remain localized and are not dominant. The majority of predictions exhibit low error (blue), confirming consistent accuracy across the workspace. These results indicate that the multi-input architecture, when individually optimized, maintains strong adaptability even under degraded conditions.

4.3 Robustness Comparison under AP Failure Conditions

This section contrasts the two proposed architectures under AP failure scenarios. The single-input DNN

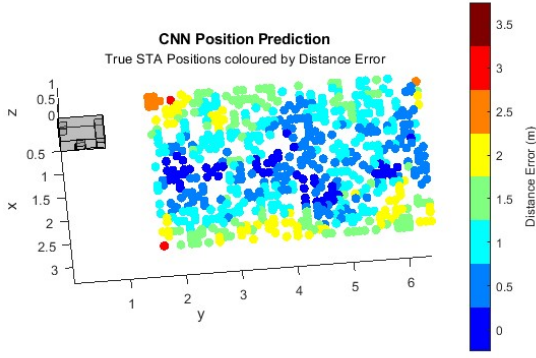


Fig. 14: Localization errors under AP1 failure with optimized AP placement (Solution 2, configuration 3).

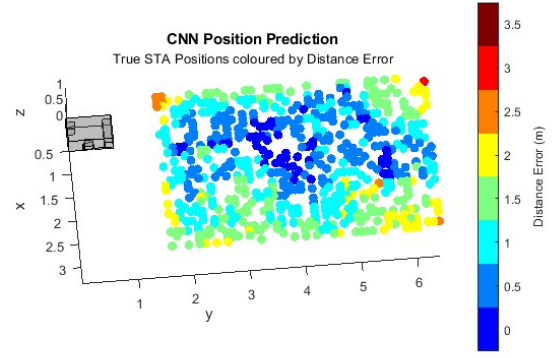


Fig. 15: Localization errors under AP2 failure with optimized AP placement (Solution 2, configuration 3).

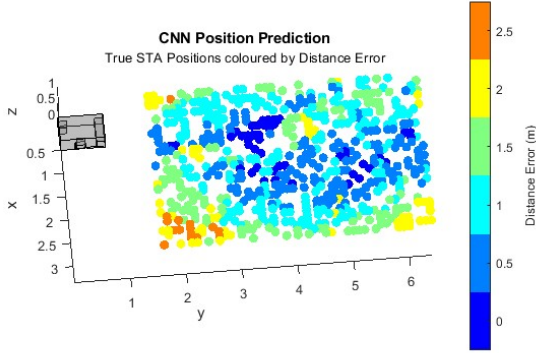


Fig. 16: Localization errors under AP3 failure with optimized AP placement (Solution 2, configuration 3).

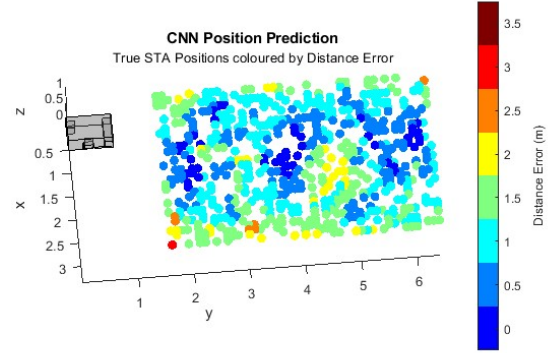


Fig. 17: Localization errors under AP4 failure with optimized AP placement (Solution 2, configuration 3).

performs well under ideal conditions ($ADE = 0.552$ m) but degrades sharply when an AP fails. Meanwhile, the multi-input DNN—especially its third configuration with branch-wise hyperparameter tuning—shows improved robustness, maintaining ADEs between 1.15 and 1.19 m. While Solution 1 is lightweight and suitable for constrained devices, it lacks adaptability. In contrast, Solution 2 introduces structural flexibility via dedicated AP branches and benefits from enhanced convergence stability and accuracy. Its consistent performance under degraded conditions makes it preferable for dynamic or fault-prone environments. The choice depends on application needs—favoring either efficiency or resilience. From a computational perspective, this improved robustness comes at the cost of increased inference latency. Specifically, the single-input DNN achieves an average inference time of 19.32 ms per location, whereas the proposed multi-input architecture requires 54.62 ms per location due to its parallel branch structure and multi-AP feature aggregation. Nevertheless, the resulting update rate of approximately 18 Hz remains compatible with near real-time indoor positioning and mobile robotic applications, where localization and planning modules typically operate at frequencies of 10–20 Hz.

Overall, the comparative analysis indicates that while multi-input DNN architectures substantially improve robustness against AP failures, spatial degradation near map boundaries remains an inherent physical challenge. This limitation is driven less by network design and more by unfavorable AP geometry and propagation constraints in these zones, underscoring the critical need for deployment-aware optimization in future fault-tolerant indoor localization systems.

4.4 Comparison with Baselines and Existing Studies

Classical indoor localization methods such as trilateration, ToA, and AoA rely on strict geometric constraints and require at least three well-distributed APs to produce a unique 2D position estimate. Under AP outage scenarios, these conditions are violated, leading to infeasible solutions or severe geometric dilution of precision (GDOP), particularly in corner and boundary regions [2]. Moreover, in rich multipath environments, reflections are typically treated as noise, which further degrades performance.

Deep learning approaches based on CIR fingerprinting alleviate these limitations by learning complex propa-

gation characteristics directly from data. Early CNN-based CIR localization frameworks [20] demonstrated improved accuracy under nominal conditions but generally assumed stable infrastructure with full AP availability. Subsequent fault-tolerant models, including RNN-based approaches [22], exploit temporal dependencies to mitigate missing inputs, at the expense of increased inference latency. More recent attention-based architectures, such as the multi-domain Deep Attention Network (DAN) [23], further enhance accuracy through feature-domain fusion but introduce substantially higher model complexity.

In contrast, the proposed framework focuses on structural robustness to AP failures through a branch-specific multi-input architecture. Unlike ensemble methods that combine independent predictions, feature-level fusion is performed within a single network, enabling cross-AP compensation prior to regression. Furthermore, Hyperband-driven branch-wise optimization allows each AP branch to adopt an appropriate capacity under failure conditions. As a result, the proposed multi-input model maintains consistent localization accuracy (ADE of 1.15–1.19 m) across all AP outage scenarios, outperforming both the single-input baseline and prior CIR-based studies that report ADEs above 1.0 m under nominal settings [24–26].

5. CONCLUSION

This paper investigated the robustness of CIR-based indoor localization under AP failure conditions using deep neural networks. Two architectures were studied: a single-input DNN that aggregates CIR measurements into a unified representation, and a multi-input DNN with dedicated processing branches for each AP. The single-input DNN architecture achieved the highest localization accuracy under normal operating conditions, delivering an ADE of 0.552 m when all APs were available. However, its performance degraded significantly under AP failure scenarios, with ADE values increasing to the range of 0.776–1.019 m. This degradation arises from the architecture's strong dependency on a complete CIR input structure, where missing or corrupted AP signals disrupt the global feature representation. In contrast, the multi-input architecture, particularly when combined with branch-specific hyperparameter optimization, demonstrated superior fault tolerance. It maintained consistently stable performance across all AP failure scenarios, with ADE values confined to a narrow range of 1.15–1.19 m. This robustness is attributed to its modular design, which allows each AP branch to be processed independently while enabling the network to adapt dynamically to partial AP availability. From a deployment perspective, the single-input architecture offers higher computational efficiency and is therefore suitable for stable environments with limited computational resources. The multi-input architecture incurs higher computational cost due to its increased structural complexity but provides significantly improved

resilience to infrastructure failures. As a result, it is better suited for mission-critical or fault-prone environments where maintaining localization reliability is paramount.

Overall, the results highlight a clear trade-off between computational efficiency and fault tolerance, providing practical guidelines for selecting appropriate DNN architectures based on application requirements and infrastructure reliability. Future work will prioritize bridging the gap between simulation-based studies and real-world deployment. Specifically, to validate the accuracy of current MATLAB-based ray-tracing and AWGN models, a physical testbed using IEEE 802.11az hardware will be developed. This setup will evaluate how well the models capture real-world interference and dynamic shadowing effects. In addition, Sim-to-Real transfer learning strategies will be investigated to efficiently adapt DNN models pre-trained in simulation to physical environments, ensuring that the proposed multi-input architecture maintains its robustness even under practical random AP failure conditions.

ACKNOWLEDGMENT

This work is funded by Vietnam - Czech bilateral project “NEO classification of signals (NEOCLASSIG) for radiosurveillance systems” under grant number NDT/CZ/22/12.

The authors would like to thank Dr. Shaik Mohammed Salman of Hitachi Energy AB for his valuable discussions and insights.

DECLARATION

The authors gratefully acknowledge the assistance of ChatGPT (OpenAI) for supporting grammar checking of manuscript writing.

CONFLICT OF INTEREST

The authors declare that they have no conflict of interest.

REFERENCES

- [1] C. Wu, Z. Yang, and Y. Liu, *Wireless Indoor Localization*. Berlin, Germany: Springer, 2018.
- [2] F. Zafari, A. Gkelias, and K. K. Leung, “A survey of indoor localization systems and technologies,” *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, Aug. 2019.
- [3] Y. Yang *et al.*, “Positioning using wireless networks: Applications, recent progress and future challenges,” *IEEE Journal on Selected Areas in Communications*, 2024.
- [4] H. A. Obeidat *et al.*, “Indoor localization using received signal strength,” in *Proc. 8th IEEE Design and Test Symposium (DTS)*, Abu Dhabi, UAE, 2013, pp. 1–6.
- [5] K. Namee *et al.*, “Improving indoor navigation accuracy with neural networks: A focus on signal propagation challenges,” in *Proc. 13th Int. Conf. on*

- Networks, Communication and Computing*, Tokyo, Japan, 2025, pp. 32–37.
- [6] Y. Chapre, P. Mohapatra, S. Jha, and A. Seneviratne, “Received signal strength indicator and its analysis in a typical WLAN system,” in *Proc. 38th Annual IEEE Conf. on Local Computer Networks (LCN)*, Sydney, Australia, 2013, pp. 304–307.
 - [7] Z. Yang, Z. Zhou, and Y. Liu, “From RSSI to CSI: Indoor localization via channel response,” *ACM Computing Surveys*, vol. 46, no. 2, pp. 1–32, Dec. 2013.
 - [8] W. Zhao *et al.*, “Crowdsourcing and multisource fusion-based fingerprint sensing in smartphone localization,” *IEEE Sensors Journal*, vol. 18, no. 8, pp. 3236–3247, Apr. 2018.
 - [9] Y. Wang *et al.*, “Bluetooth positioning using RSSI and triangulation methods,” in *Proc. IEEE Consumer Communications and Networking Conf. (CCNC)*, Las Vegas, NV, USA, 2012, pp. 837–842.
 - [10] B. Jin, X. Xu, and T. Zhang, “Robust time-difference-of-arrival (TDOA) localization using weighted least squares with cone tangent plane constraint,” *Sensors*, vol. 18, no. 3, pp. 1–12, Mar. 2018.
 - [11] L. C. Godara, “Application of antenna arrays to mobile communications. II. Beam-forming and direction-of-arrival considerations,” *Proc. IEEE*, vol. 85, no. 8, pp. 1195–1245, Aug. 1997.
 - [12] M. I. AlHajri, N. T. Ali, and R. M. Shubair, “Indoor localization for IoT using adaptive feature selection: A cascaded machine learning approach,” *IEEE Antennas Wireless Propag. Lett.*, vol. 18, no. 11, pp. 2306–2310, Nov. 2019.
 - [13] S. Maranò, W. M. Gifford, H. Wymeersch, and M. Z. Win, “NLOS identification and mitigation for localization based on UWB experimental data,” *IEEE J. Sel. Areas Commun.*, vol. 28, no. 7, pp. 1026–1035, Sep. 2010.
 - [14] M. T. Hoang *et al.*, “Recurrent neural networks for accurate RSSI indoor localization,” *IEEE Internet Things J.*, vol. 6, no. 6, pp. 10639–10651, Dec. 2019.
 - [15] X. Wang, X. Wang, and S. Mao, “Deep convolutional neural networks for indoor localization with CSI images,” *IEEE Trans. Netw. Sci. Eng.*, vol. 7, no. 1, pp. 316–327, Jan. 2020.
 - [16] M. Cai and Z. Lin, “Precise WiFi indoor positioning using deep learning algorithms,” arXiv:2307.02011, 2023.
 - [17] K. Liu *et al.*, “Toward low-overhead fingerprint-based indoor localization via transfer learning: Design, implementation, and evaluation,” *IEEE Trans. Ind. Informat.*, vol. 14, no. 3, pp. 898–908, Mar. 2018.
 - [18] R. Vishwakarma, R. B. Joshi, and S. Mishra, “IndoorGNN: A graph neural network based approach for indoor localization using WiFi RSSI,” in *Proc. Int. Conf. Big Data Analytics*, Jaipur, India, 2023, pp. 150–165.
 - [19] Z. Yun and M. F. Iskander, “Ray tracing for radio propagation modeling: Principles and applications,” *IEEE Access*, vol. 3, pp. 1089–1100, Jul. 2015.
 - [20] V.-L. Dao and S. M. Salman, “Deep neural network for indoor positioning based on channel impulse response,” in *Proc. IEEE 27th Int. Conf. Emerging Technologies and Factory Automation (ETFA)*, Stuttgart, Germany, 2022, pp. 1–8.
 - [21] *IEEE Draft Standard for Information Technology—Telecommunications and Information Exchange Between Systems—Local and Metropolitan Area Networks*, IEEE P802.11be/D3.0, Jan. 2023.
 - [22] S. Huang, F. Liao, and R. S. H. Teo, “Fault tolerant control of quadrotor based on sensor fault diagnosis and recovery information,” *Machines*, vol. 10, no. 11, p. 1088, Nov. 2022.
 - [23] P. Susarla *et al.*, “Indoor positioning with multi-domain CSI-based deep attention networks for MIMO wireless systems,” *Research Square*, preprint, Jul. 2025, doi: 10.21203/rs.3.rs-6817871/v1.
 - [24] D. K. Nguyen, L. C. Nguyen, and M. K. Hoang, “A Bayesian optimization based deep learning model for Wi-Fi fingerprinting based indoor positioning,” *J. Commun.*, vol. 19, no. 10, pp. 1–12, 2024.
 - [25] M. Chen and Q. Pu, “Bayesian optimized indoor positioning algorithm based on dual clustering,” *Sci. Rep.*, vol. 15, pp. 1–12, Mar. 2025.
 - [26] H. Cao *et al.*, “LOS compensation and trusted NLOS recognition assisted WiFi RTT indoor positioning algorithm,” *Expert Syst. Appl.*, vol. 243, p. 122867, 2024.