INTERNATIONAL JOURNAL OF COMPUTERS COMMUNICATIONS & CONTROL
Online ISSN 1841-9844, ISSN-L 1841-9836, Volume: 21, Issue: 2, Month: April, Year: 2026
Article Number: 7398, https://doi.org/10.15837/ijccc.2026.2.7398

CCC Publications

AGORA
UNIVERSITY PRESS

High-Accuracy Indoor Positioning Using Deep Neural Networks Combining with Optimal Hyperparameters and Fault-Tolerant Mechanism

V.-P. Hoang, T.-K.-L. Ta, V.-L. Dao, T.-Y. Hoang

Van-Phuc Hoang, Thi-Kieu-Lan Ta, Thi-Yen Hoang*
Le Quy Don Technical University
Hanoi, Vietnam
phuchv@lqdtu.edu.vn, lanmtak54@gmail.com
*Corresponding author: yenht@lqdtu.edu.vn

Van-Lan Dao
Hitachi Energy AB
Sweden
Lan.dao@hitachienergy.com

Abstract

High-accuracy indoor positioning is essential for reliable wireless services and drone monitoring, yet the performance remains highly sensitive to access point (AP) failures. This paper proposes a fingerprinting-based localization framework utilizing Channel Impulse Response (CIR) features and a deep convolutional neural network (DCNN) to ensure robust positioning under both nominal and degraded operational conditions. Two scenarios are examined. In the first scenario, where all four APs operate normally, the influence of network depth and hyperparameter settings on localization accuracy is systematically evaluated. The network block count is progressively reduced and optimal hyperparameters are selected via the Hyperband algorithm, followed by parameter refinement through simulated annealing in a two-stage optimization strategy. Second, in single-AP failure conditions, a mitigation approach is employed wherein a single-input DCNN is retrained for each failure case, and its hyperparameters are independently optimized using Hyperband to compensate for reduced signal diversity. Experimental validation demonstrates that the proposed system achieves an average distance error (ADE) of 0.552 m under nominal conditions, outperforming existing methods. Under single-AP failure scenarios, the framework maintains strong positioning performance, with ADE ranging from 0.776 m to 1.019 m. These results confirm the effectiveness and resilience of the proposed DCNN architecture, highlighting its suitability for reliable indoor localization in realistic and fault-prone environments.

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

1 Introduction

Indoor positioning systems (IPS) have attracted significant attention due to their diverse applications in domains such as factory automation, mining, and warehouse management [1]. In industrial settings, mobile robots rely on IPS to transport items between stations equipped with specialized machinery, enabling real-time task allocation and optimizing travel paths to enhance operational throughput. In underground mining environments, autonomous robots depend on IPS for navigation and safety, with stringent regulatory requirements, such as those outlined in the MINER Act [2], which require precise tracking of personnel and assets. Similarly, in warehouse operations, IPS facilitates efficient item retrieval and shelf transportation tasks [3], reducing the number of active robots needed while maximizing overall efficiency. Especially, it is also important for indoor drone positioning in drone monitoring systems [4].

Unlike outdoor positioning, which benefits from the availability of the Global Positioning System (GPS), indoor environments pose distinct challenges, including signal obstruction, multipath propagation, and complex architectural layouts that degrade conventional positioning performance [5]. To overcome these obstacles, various technologies have been explored, from radio frequency signals and ultrasonic waves to visual markers [1]. Among these, WLAN-based fingerprinting systems are particularly attractive for their cost effectiveness and ease of deployment [6]. However, traditional fingerprinting approaches often suffer from high-dimensional data variability and environmental dynamics, leading to suboptimal localization accuracy [7]. Recent advances in deep neural networks (DNNs) have shown considerable promise in extracting intricate spatial features from complex data sets and improving the robustness of IPS [8].

Building on previous work that optimized the number and placement of access points (APs) [9], this study focuses on further enhancing IPS performance through DCNN-based modeling. Specifically, Channel Impulse Response (CIR) data is used as input features. To optimize model performance, hyperparameter tuning is conducted using the Hyperband algorithm, while Simulated Annealing (SA) is employed to refine AP placement efficiently. Furthermore, to address real-world deployment challenges, the work investigates fault-tolerant mechanisms by handling scenarios where an AP experiences failure. Solutions such as robust hyperparameter pre-selection and network adaptation are proposed to sustain high localization accuracy despite AP outages. Hyperband and SA have been introduced and extensively studied in the literature for hyperparameter and parameter optimization tasks [10]. These algorithms are widely adopted in various machine learning applications due to their efficiency in navigating large search spaces and avoiding local minima. However, to the best of our knowledge, no prior work has explored the combination of Hyperband and SA in the context of IPS. This motivates the development of our proposed hybrid optimization framework, which aims to leverage the complementary strengths of both algorithms for enhanced localization performance.

While several deep learning based indoor localization works have reported promising accuracy, most existing approaches rely on RSSI or RSS features and do not explicitly consider system robustness under infrastructure failures. Specifically, Nabati and Ghorashi [8] achieved a mean positioning error of 0.93 m using a DNN augmented with previous position estimates; however, their framework assumes continuous AP availability and does not incorporate systematic hyperparameter optimization. Nguyen et al. [32] applied Bayesian optimization to tune LSTM hyperparameters, but their method focuses on network tuning only and does not address optimal AP deployment or fault-tolerant operation.

In contrast, the proposed approach adopts CIR features, which preserve richer physical-layer characteristics than RSSI-based measurements, and jointly optimizes both DCNN hyperparameters and AP placement through a hybrid Hyperband-and-SA framework. Furthermore, to the best of our knowledge, no prior work has simultaneously addressed hyperparameter optimization, AP placement optimization, and fault-tolerant positioning within a unified deep learning based indoor localization framework. These aspects constitute the main contributions of this study.

The remainder of this paper is organized as follows. Section 2 reviews related works of indoor positioning systems and deep learning-based localization approaches. Section 3 presents the proposed method, including the formulation of CIRs, the deep convolutional neural network (DCNN) model, the optimization strategies based on the Hyperband and SA algorithms, and the evaluation methodology. Section 4 presents the performance evaluation results under two distinct operational scenarios: (i) nominal conditions with all four APs functioning optimally, accompanied by a comprehensive comparative analysis of various DNN architectural configurations, and (ii) degraded operational conditions characterized by single AP failure. Finally, Section 5 concludes this work and discusses the future research directions.

2 Related Work

Indoor localization has attracted considerable research interest due to its critical role in smart buildings, healthcare, and robotics applications. Traditional methods, such as Received Signal Strength Indicator (RSSI)-based techniques, are favored for their hardware simplicity and ease of deployment [11]. However, RSSI approaches suffer from limited positioning accuracy, vulnerability to signal obstructions, low Signal-to-Noise Ratio (SNR), and susceptibility to Non-Line-of-Sight (NLoS) propagation effects [12]. More advanced signal-based techniques, including Channel State Information (CSI), provide richer data representations, capturing both amplitude and phase information, thus enabling higher localization precision [13]. Nonetheless, CSI-based methods often introduce increased system complexity and require specialized hardware compared to RSSI-based approaches. Indoor localization techniques can broadly be categorized into signal-based, time-based, and angle-based methodologies. Signal-based fingerprinting methods, particularly those utilizing RSSI, are cost-effective and leverage existing wireless infrastructures, such as Wi-Fi and Bluetooth [14, 15]. While model-based and fingerprinting approaches exist, fingerprinting tends to exhibit greater robustness under multipath and dynamic environments [16]. Time-based techniques, including Time of Arrival (ToA), Time Difference of Arrival (TDoA), and Round-trip Time of Flight (RToF), improve localization performance under Line-of-Sight (LoS) conditions, but require strict synchronization and high hardware complexity [17, 18]. Angle-based methods, such as Angle of Arrival (AoA) and Phase of Arrival (PoA), infer position from signal direction or phase difference, offering high accuracy with fewer anchors, but requiring specialized directional antennas and remaining sensitive to multipath propagation [19].

Machine learning (ML) techniques have been explored to overcome the limitations of traditional methods. K-Nearest Neighbor (KNN) is widely used for its simplicity but struggles with scalability and sensitivity to noise [20]. Support Vector Machines (SVMs) and their variants, including the least squares support vector machine classifier (LS-SVMC) [21], the novel normalized rank-based support vector machine classifier (NR-SVM) [22], and the online independent support vector machine (OISVM) [23], have demonstrated improved robustness in high-dimensional and dynamic environments [24]. Neural network models, such as Backpropagation Neural Networks (BPNN) and Radial Basis Function Networks (RBFN), have also been applied to RSSI data [25], though their scalability remains limited compared to recent deep learning (DL) approaches.

Deep learning has emerged as a transformative tool for indoor localization by extracting complex features from noisy signal environments. Recurrent Neural Networks (RNNs) have been shown to effectively capture user motion and temporal dependencies in trajectory-based localization [26]. Nabati et al. [8] enhanced real-time fingerprinting by integrating previous user states into a DNN model. Hybrid models combining different input types, such as convolutional neural networks (CNNs) integrating RSSI and AoA [27], have achieved sub-meter accuracy in cluttered environments. DCNN using CSI phase data has further improved angle estimation accuracy over RSSI-only methods [28]. Deep Belief Networks (DBNs) have been leveraged to reduce reliance on labeled datasets [29], while transfer learning has enhanced generalization across diverse environments [30]. Graph-based approaches, such as IndoorGNN [31], have reformulated localization as a classification task on dynamic graphs, achieving over 95% accuracy on public datasets. Bayesian Optimization frameworks have been applied to tune hyperparameters for Long Short-Term Memory (LSTM)-based models, demonstrating notable improvements in mean distance error (MDE) when combined with feature reduction techniques like Singular Value Decomposition (SVD) and Linear Discriminant Analysis (LDA) [32].

While significant progress has been made in enhancing localization accuracy, most existing works assume a fixed number and placement of APs, overlooking the impact of AP failures which is a critical factor in real-world deployments. Furthermore, many studies primarily focus on improving signal modeling without addressing model robustness to infrastructure disruptions. This study aims

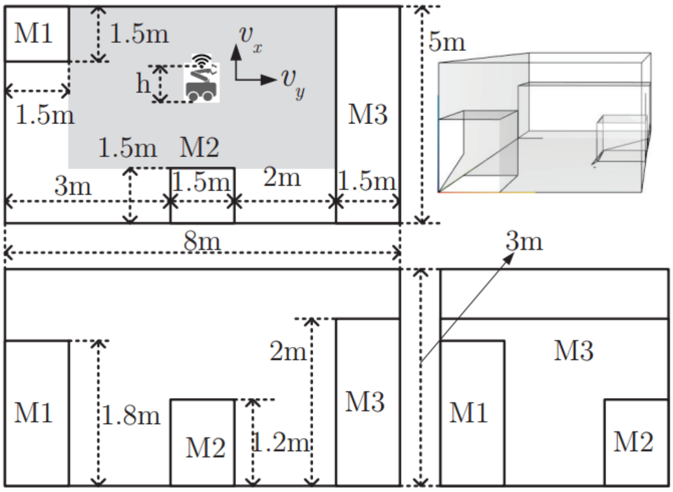

Figure 1: Illustration of an indoor environment where four APs communicate with a mobile robot

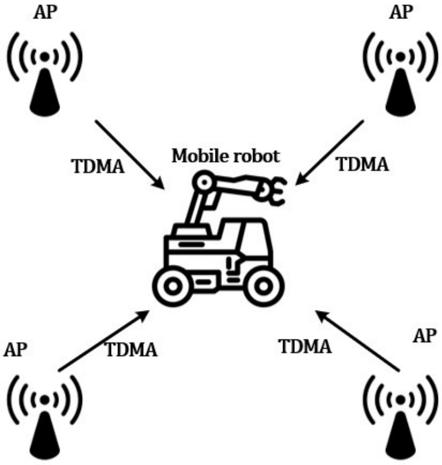

Figure 2: TDMA-based downlink and simultaneous uplink communication between APs and the mobile robot

to bridge these gaps by proposing a DCNN-based optimization framework that not only enhances localization accuracy through hyperparameter and parameter tuning but also maintains performance resilience under AP failure scenarios. By focusing on model-level adaptation rather than network-level redesign, the proposed approach offers a scalable and practical solution for reliable indoor localization in dynamic industrial environments.

3 Proposed Method

A system comprising four APs communicating with a mobile robot (or a drone in a specific scenario) in an indoor environment is considered, as depicted in Figure 1, where the ROI of 5×3.5 m^2 is partitioned into four sub-regions, each assigned to one AP. The robot is constrained to move on a horizontal plane with coordinates (x_m, y_m), while the APs are stationary at fixed positions (x_i, y_i, z_i). To enable self-localization, the robot exchanges signals with all four APs via downlink and uplink communications. In the downlink, a Time Division Multiple Access (TDMA) scheme is employed to prevent signal interference between APs, as shown in Figure 2, where each AP transmits sequentially in the downlink while the robot broadcasts simultaneously in the uplink. By analyzing the strengths of the received signal, the robot estimates its position. In the uplink, the robot broadcasts signals that are received by all APs simultaneously.

Assuming full connectivity among APs, the system collaboratively estimates the robot's position, enabling functions such as optimal path planning. For accurate channel modeling, the ray tracing algorithm [33] is adopted due to its ability to simulate electromagnetic wave propagation over a wide frequency range (100 MHz – 100 GHz), encompassing the operating frequency of the APs (802.11az Wi-Fi standard). Unlike simplified propagation models that primarily consider direct paths, ray tracing accounts for complex multipath phenomena such as reflections, diffractions, and scatterings prevalent in indoor environments. It computes the path loss and phase shift for each ray using Fresnel equations, the Uniform Theory of Diffraction, and material-specific properties. This method, implemented through simulation platform (MATLAB 2022a), provides a detailed and realistic 3D propagation model.

In the proposed method, the data generation process and CIR construction are presented first. The obtained CIRs are then used as input features for the DCNN model. The overall workflow of the proposed framework is summarized in Figure 3. It illustrates a two-stage localization process. In Stage 1, the DCNN model is optimized using the Hyperband algorithm to select the best hyperparameters based on the average distance error (ADE). Subsequently, the SA algorithm is applied to optimize the placement of APs, guided by the performance of the DCNN. Once the ADE falls below a predefined

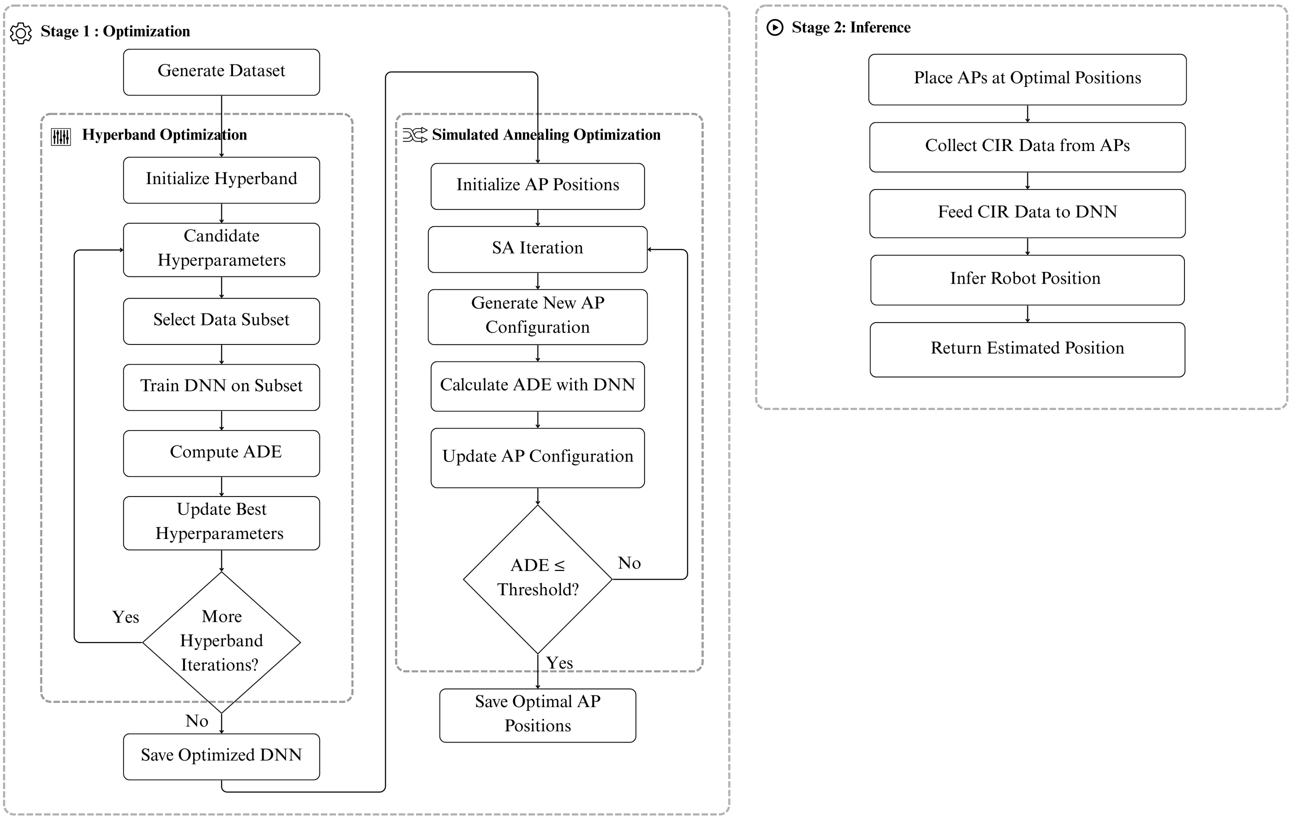

Figure 3: Diagram of the proposed method using Hyperband and Simulated Annealing for DNN optimization and AP deployment evaluation

threshold, the optimal AP configuration is stored. In Stage 2, for each AP position, the corresponding CIR values are simulated and collected, then fed into the trained DNN to estimate the robot's location. The final output is the estimated position of the robot. Subsequent parts will provide the details of these components and show the ADE metric for evaluating the localization performance.

Various types of wireless channel signatures have been investigated in prior studies, including RSSI, CSI, and CIR. Among these, CIR offers greater potential for enhancing system performance, particularly regarding physical-layer security. In this study, CIR is selected as the input feature for the DCNN model due to its ability to capture rich environmental characteristics and its uniqueness for each wireless node pair [34]. The CIR is measured for each AP-to-robot link and then used as the input to the DCNN for localization purposes. The CIR can be mathematically expressed as (1).

$$\mathrm{CIR}(t) = \sum_{i=1}^{N} h_i \left[W(t - \tau - t_i) \right] \tag{1}$$

where W, N, h_i, τ, and t_i denote the bandwidth, number of paths, complex channel gain of the i-th path, propagation delay, and delay spread between the first and i-th paths, respectively.

The indoor environment layout employed in this paper is illustrated in Figure 1, adapted from our previous work [9]. The robot moves within a defined region of interest (ROI) measuring 5 × 3.5 m^2, which is partitioned into four sub-regions, each containing one AP to ensure full coverage. The process of acquiring CIR measurements between each AP and Station (STA) pair is illustrated in Figure 4. The 802.11az Wi-Fi standard [35] is adopted for packet transmission. Each AP transmits an 802.11az packet over an Additive White Gaussian Noise (AWGN) channel, and each STA receives the transmitted signals. It is assumed that STAs can distinguish between different APs, and that there is no inter-AP interference. Packet reception at a specific location is deemed unsuccessful if the received SNR falls below a given threshold, resulting in a zero-vector CIR. In this work, SNR values of 20 dB, 25 dB, and 30 dB are investigated.

Key system parameters include a bandwidth of $W = 160$ MHz and a robot movement velocity of $(v_x, v_y) = (1, 1)$ m/s. The APs are positioned at coordinates $(x_i, y_i, 2.9)$ meters (m). The collected dataset is divided into three subsets: a training set (80%), a validation set (10%), and a testing set (10%). The training and validation sets are used during the model optimization phase, while the test set is reserved exclusively for evaluating the final localization performance. Table 1 summarizes the main simulation and training parameters used in this study, including the antenna configuration, channel bandwidth, SNR assumptions, geometric layout, and optimization settings.

Table 1: Main simulation and training parameters

Parameter	**Value**	**Parameter**	**Value**
Antenna array (TX/RX)	4×1 Uniform Linear Array (ULA)	STA spacing	0.1 m
Surface material	Metal (worst-case reflective environment)	Signal bandwidth	160 MHz
Guard interval	1.6 μs	SNR levels	20, 25, 30 dB
ROI size	5×3.5 m^2	AP height	2.9 m
SA initial temperature (T)	2	SA cooling factor (ϵ)	0.95
SA iterations (L)	200	Mini-batch size	128
L2 regularization	1×10^{-4}	LR drop period / factor	30 epochs / 0.1
Sub-region partitioning	4 horizontal bands	Software Environment	MATLAB 2022a

In our previous study [9], we proposed a DCNN architecture consisting of K stacked blocks. Each block comprises a convolutional layer (Conv), a batch normalization layer (BatchNorm), a rectified linear unit (ReLU) activation, a pooling layer (Pool), and a dropout layer. Following these blocks, a fully connected (FC) layer and a regression layer are employed for position estimation, as given in Figure 5, where K convolutional blocks are stacked sequentially.

The convolutional layers extract features from the input data using trainable kernels, and stacking multiple Conv layers enables the network to capture increasingly complex and abstract representations. Batch normalization and ReLU activation are incorporated to accelerate the training process and improve network stability. To mitigate overfitting, a dropout layer randomly deactivates neurons during training. Meanwhile, pooling layers downsample the feature maps, reducing spatial dimensions and computational cost while preserving essential information. However, in our earlier work, several architectural parameters, such as the number of blocks $K = 7$ and the number of filters per convolutional layer (fixed at 256), were manually set and remained constant across experiments. In this study, we aim to optimize the DCNN by exploring the impact of different hyperparameter configurations, including the number of blocks and other key parameters, to achieve a balance between computational efficiency and localization accuracy. The proposed DCNN structure, depicted in Figure 5, consists of stacked convolutional blocks. The hyperparameter search space is defined as follows. The Filters is with 32, 64, 128, 256, 512; Dropout rate is 0.1, 0.2, 0.3, 0.4, 0.5; and learning rate of 1e-2, 1e-3, 1e-4. Each configuration is trained for 30 epochs, with performance evaluated by the Hyperband al-

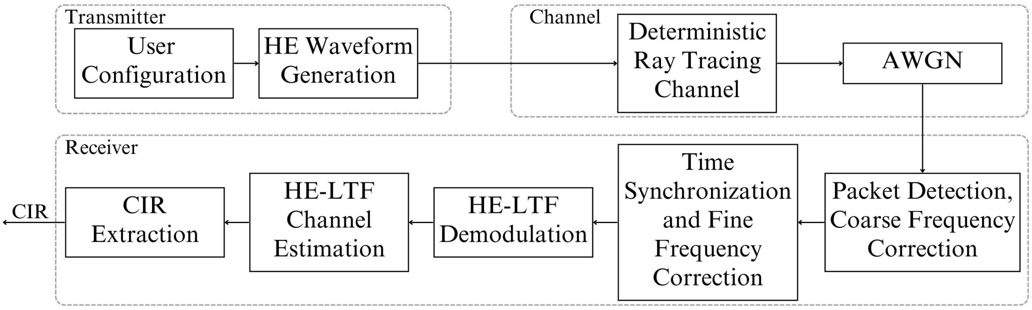
Transmitter
User Configuration
HE Waveform Generation
Channel
Deterministic Ray Tracing Channel
AWGN
Receiver
CIR
CIR Extraction
HE-LTF Channel Estimation
HE-LTF Demodulation
Time Synchronization and Fine Frequency Correction
Packet Detection, Coarse Frequency Correction

Figure 4: CIR acquisition process

gorithm after every 10 epochs. These ranges are selected based on standard architectural conventions and empirical considerations. The filter sizes follow the power-of-two scaling commonly used in CNN architectures (VGG and ResNet) to balance model capacity and computational cost. The dropout upper bound of 0.5 is chosen according to established regularization practice, while the learning rate candidates are centered around the default value of the Adam optimizer to ensure stable convergence within the 30-epoch training budget.

To determine the optimal hyperparameters for each network, the Hyperband algorithm [10] is employed. Hyperband accelerates the evaluation of each configuration by: (1) Successive Halving, in which the hyperparameter search space is partitioned into multiple brackets, each representing different allocations of models and training resources; (2) Early Stopping, whereby training is terminated if the loss function shows insufficient improvement over a set number of epochs, enabling the algorithm to discard less promising models early and concentrate resources on better-performing candidates; and (3) Balancing Exploration and Exploitation, where the algorithm initially explores a wide variety of configurations to locate promising regions and subsequently allocates more resources to exploit these regions through further model training.

The Hyperband algorithm, as presented in Algorithm 1, takes as input R, the maximum number of training epochs allocated to a single hyperparameter configuration, and η, the halving factor determining the proportion of configurations discarded in each `SUCCESSIVEHALVING` round. B denotes the total resource budget, and $s_{\max}$ represents the maximum number of successive halving iterations. In each outer iteration, n hyperparameter configurations, denoted as **config_set**, are generated. Each configuration **config** is trained for r epochs, and its performance is evaluated by computing the validation ADE, collected into **ADE_set**. A fraction of the worst-performing configurations is then discarded based on the ADE values. The values of n and r are subsequently updated for the next halving iteration.

To identify the minimum number of APs required, along with their optimal mounting positions, while satisfying the given performance metric constraint (ADE $\leqslant$ ADE$_{threshold}$), Simulated Annealing Algorithm is utilized in this stage. To ensure practical deployment, the height of the APs is fixed and their positions are restricted to accessible areas within the indoor environment. Our objective is to determine the optimal positions for the APs at $(x_{n,i}^{\text{opt}}, y_{n,i}^{\text{opt}})$, ensuring robust coverage and minimized error. The proposed algorithm employs a SA, as presented in Algorithm 2, processed to determine the optimal AP placement. The region of interest is divided into n equal sub-areas as shown in Figure 6, where each sub-area defines the initial search region for its corresponding AP, ensuring full spatial

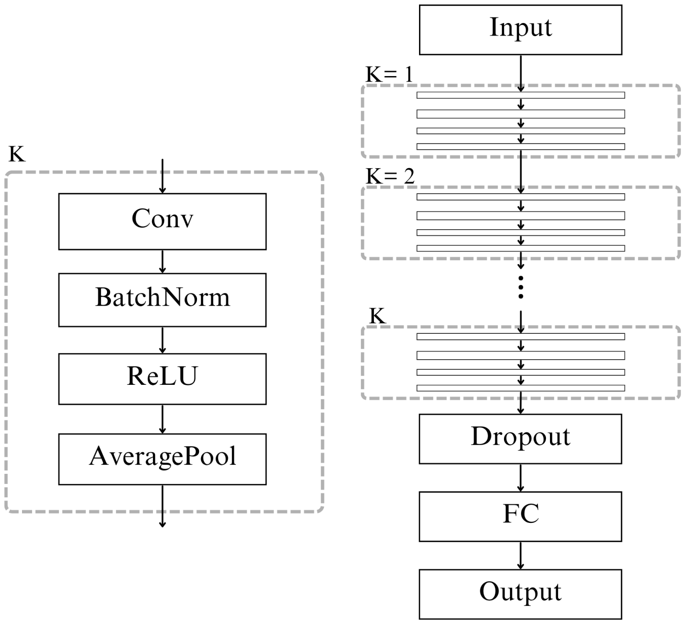

Figure 5: Architecture of the proposed DCNN with K sequential convolutional blocks for position estimation

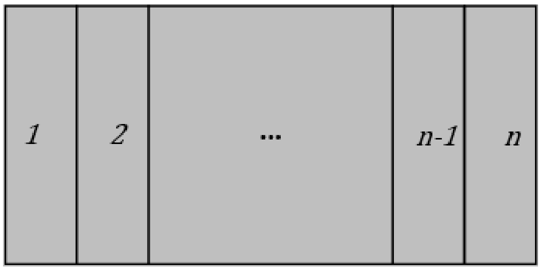

Figure 6: Partition the gray area into n equal sub-areas

Algorithm 1: Hyperband Algorithm for Hyperparameter Optimization

```
Input: $R, \eta$ (default: $\eta = 3$)
Initialization: $s_{\max} = \lfloor \log_\eta(R) \rfloor$, $B = (s_{\max} + 1)R$
for $s \in \{s_{\max}, s_{\max} - 1, \dots, 0\}$ do
    $n = \lceil \frac{B}{\eta^{s+1}} \rceil$, $r = R\eta^{-s}$
    // begin SUCCESSIVEHALVING
    config__set = get_config($n$)
    for $i \in \{0, \dots, s\}$ do
        $n_i = \lfloor n\eta^{-i} \rfloor$
        $r_i = r\eta^i$
        ADE__set = {train_return_ADE($\mathbf{c}, r_i$) :
            $\mathbf{c} \in$ config__set}
        config__set = select_config(config__set,
            ADE__set, $\lfloor n_i/\eta \rfloor$)
    end for
end for
```

Algorithm 2: Finding minimum number of APs and their optimal positions

```
Input: ADE$_{\text{threshold}}$
Output: ADE$_{\min}$, APs ($n$), ($x_{\text{opt}}, y_{\text{opt}}$)
function MAIN
    Num APs: $n \leftarrow n_0$
    while true do
        Partition area into $n$ sub-areas
        Init solution $S_0 = (x_0, y_0)$
        Calc cost at $S_0$: ADE
        for $i \leftarrow 1$ to $L$ do
            Neighbor $S' = (x_i, y_i)$
            Calc cost at $S'$: ADE
            $\Delta$ = ADE $-$ ADE$_0$
            if $\Delta < 0$ then
                $S \leftarrow S'$, ADE$_{\min} \leftarrow$ ADE
            else
                Prob $\delta = e^{-\frac{\Delta}{T}}$
                if rand$[0, 1) < \delta$ then
                    $S \leftarrow S'$
                    ADE$_{\min} \leftarrow$ ADE
                end if
            end if
            Temp: $T \leftarrow \epsilon T$
        end for
        if ADE$_{\min} \leq$ ADE$_{\text{thresh}}$ then
            Update $(x_{\text{opt}}, y_{\text{opt}})$
            break
        else
            $n \leftarrow n + 1$
        end if
    end while
    return ADE$_{\min}, n, (x_{\text{opt}}, y_{\text{opt}})$
end function
```

coverage at the start of the SA optimization process. The algorithm iteratively adjusts the number and positions of APs to minimize ADE, with the constraint that (ADE $\leqslant$ ADE$_{threshold}$) must be satisfied. To achieve this, a DCNN is used to predict positioning accuracy and provide feedback on ADE. For each n, the DCNN generates a new dataset tailored to the AP positions, ensuring accurate predictions for the optimization process. Key parameters in the SA process include the maximum temperature T, cooling factor ϵ, and number of iterations L. These parameters are fine-tuned experimentally to balance convergence quality and computation time. A higher number of iterations increases the likelihood of achieving a near-global optimum but comes at the cost of extended execution time. The final optimal solution is determined by running the algorithm during an offline phase, providing both the minimum number of APs and their corresponding optimal positions.

In this work, the ADE is utilized as a metric to assess the performance of the DNN model. ADE quantifies the average discrepancy between the predicted and actual positions of the object and is computed using (2).

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

where M_{test}, $(x_{m,j}, y_{m,j})$, $(\hat{x}_{m,j}, \hat{y}_{m,j})$ represents the number of test samples, the ground truth position, and the predicted position, respectively. Consequently, a lower ADE value signifies a better model performance.

4 Results

We evaluated the proposed algorithm in two scenarios. In the first scenario, all four APs are operating normally. We focus on optimizing the network parameters for the available networks and detail the hyperparameter tuning process along with the configuration of the proposed DCNN. In the second scenario, we demonstrate the robustness of the algorithm by simulating a failure in one of the APs, showing that the proposed method maintains a sufficiently low ADE.

4.1 Performance Evaluation under Full Operation of All Four APs

To demonstrate the effectiveness of the proposed method, we first present the evaluation results of existing neural network models using CIR data, followed by a comparison with our proposed DCNN model. The proposed DCNN model employs Hyperband algorithm for hyperparameters optimization and SA for network parameters optimization. We evaluate five representative DNN architectures (ResNet18, RepVGG, MobileNetV2, GoogleNet, and SqueezeNet), each with distinct structural innovations. In our investigation of these networks, we trained them using the SA algorithm to optimize the parameters, without tuning the hyperparameters.

Figure 7 presents the ADE progression over training iterations for the evaluated networks. As expected, all architectures exhibit a general trend of decreasing ADE as training progresses, indicating effective learning. However, notable differences in final performance are observed across models. After 200 iterations, ResNet18 achieves the lowest ADE of 0.585 m, followed closely by RepVGG at 0.601 m, highlighting their strong learning capability. MobileNetV2 also performs well with an ADE of 0.674 m, despite being designed for resource-constrained devices. In contrast, GoogleNet and SqueezeNet yield higher ADE values of 0.767 m and 1.044 m, respectively, suggesting limited suitability for this application. These results suggest that ResNet's residual learning mechanism facilitates deeper and more stable feature extraction. Similarly, RepVGG's streamlined structure, optimized for both training and inference, delivers competitive results. While MobileNetV2 leverages the architectural efficiency to maintain a reasonable accuracy, GoogleNet's complex multi-branch design may pose training challenges and the risk of overfitting. Although SqueezeNet is compact,

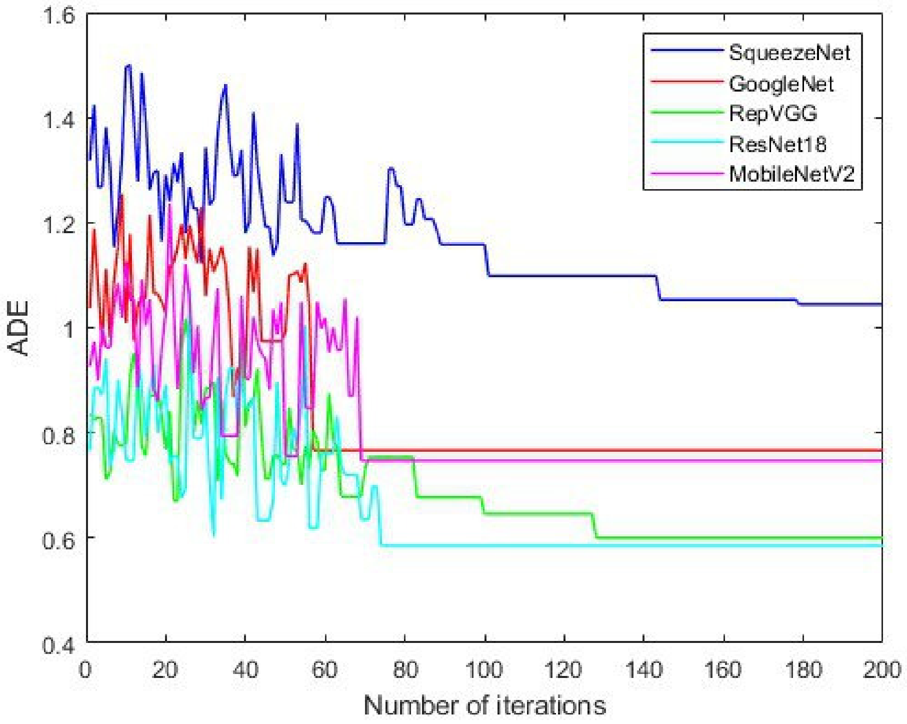

Figure 7: Impact of DNN architecture on ADE over training iterations

Table 2: Results of ADE for the model with different values of K and tuned hyperparameters after 200 iterations

Value of K	**Hyperparameter**	**Network parameters**	**ADE after 200 iterations**
$K = 8$ Hyperband	units: (64, 128, 32, 512, 128, 64, 256, 128) dropout: 0.1 learning rate: 0.001	3,558,146	0.955
$K = 7$ Not Hyperband	units: (256, 256, 256, 256, 256, 256, 256) dropout: 0.4 learning rate: 0.001	3,558,146	0.582
$K = 7$ Hyperband	units: (64, 128, 32, 256, 64, 32, 128) dropout: 0.1 learning rate: 0.0001	393,410	0.552
$K = 6$ Hyperband	units: (64, 64, 128, 64, 256, 256) dropout: 0.2 learning rate: 0.0001	929,602	0.573
$K = 5$ Hyperband	units: (256, 256, 256, 128, 128) dropout: 0.3 learning rate: 0.0001	1,637,890	0.583
$K = 4$ Hyperband	units: (256, 256, 64, 256) dropout: 0.3 learning rate: 0.0001	904,258	0.628
$K = 3$ Hyperband	units: (512, 32, 512) dropout: 0.3 learning rate: 0.0001	367,778	0.680
$K = 2$ Hyperband	units: (32, 128) dropout: 0.3 learning rate: 0.01	87,970	0.983
$K = 1$ Hyperband	units: 64 dropout: 0.3 learning rate: 0.0001	100,930	0.943

it appears to lack the representational capacity required for modeling CIR data effectively. Overall, the observed performance variations can be largely attributed to differences in architectural design, model capacity, generalization capability, and their respective suitability for capturing the specific characteristics of the localization task.

To achieve high localization accuracy, it is essential not only to select an appropriate DNN architecture, but also to carefully optimize its hyperparameters. For the proposed DCNN model, we investigate the relationship between the number of network blocks, hyperparameter configurations, and the resulting ADE. The Hyperband algorithm is employed to efficiently explore the hyperparameter space and identify optimal settings that minimize ADE and enhance model performance. Figure 8 shows the ADE convergence results of proposed DCNN model evaluation with different values of K when all four APs fully operate and Table 2 summarizes the results across varying the number of network blocks K.

As the number of blocks (K) increases, the total number of parameters also grows substantially, enabling the network to learn more complex feature representations. However, this added complexity can increase the risk of overfitting. For each value of K, the optimal hyperparameters differ, reflecting the sensitivity of the network's performance to architectural depth. Notably, for $K = 7$, hyperparameter tuning led to a noticeable improvement in ADE, approximately a 0.03 m reduction at the 200^{th} iteration. Moreover, the tuned configuration required significantly fewer parameters, approximately nine times fewer than the untuned version, highlighting the efficiency of the Hyperband approach in identifying lightweight yet effective configurations. After determining the optimal hyperparameters, the SA algorithm was applied to further minimize ADE through AP placement optimization. The results show that ADE converges to a stable value after approximately 180 iterations. Overall, ADE tends to decrease with increasing K. However, beyond $K = 4$, the marginal gains diminish. The best performance is observed at $K = 7$, achieving an ADE of 0.552 m at the 200^{th} iteration. These findings suggest that increasing network depth beyond a certain point not only adds computational

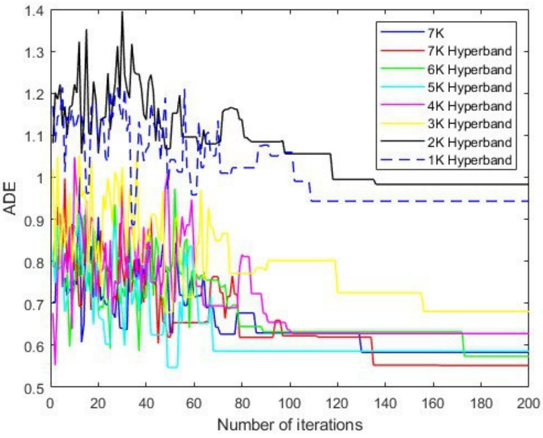

Figure 8: Results of the proposed DCNN with different K values (All APs active)

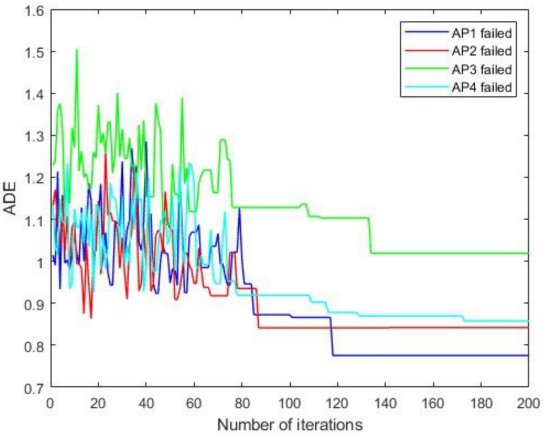

Figure 9: Results of the proposed DCNN with fixed $K = 7$ blocks (One AP fails)

complexity but may also lead to overfitting without significant accuracy improvement.

In terms of computational complexity, the optimization framework balances search thoroughness with efficiency. By employing Hyperband, the search across 75 configurations is optimized through early stopping at 10-epoch intervals. Furthermore, the resulting DCNN for $K = 7$ is highly efficient, featuring only 393,410 parameters. Since the SA-based AP placement is an offline process, the system ensures real-time responsiveness for mobile robot navigation.

4.2 Performance Results of the Situation with One AP Failure

In practical wireless environments, AP failures are inevitable and can significantly impair positioning performance. To assess the resilience of the proposed method, we examine the single-AP failure scenario and apply a tailored optimization strategy. For each AP-loss case, a dedicated hyperparameter search is conducted using the proposed DCNN architecture with $K = 7$ blocks and the Hyperband algorithm to determine the optimal network configuration. The objective is to compensate for reduced signal diversity by adapting the number of filters, dropout rate, and learning rate to the specific failure condition. Table 3 presents the network parameters and ADE results after 200 iterations, while Figure 9 illustrates the ADE progression over iterations. The results indicate that although the model can adapt under AP-failure conditions, ADE inevitably increases compared to the scenario where all four APs are operational. This highlights the system's reliance on multi-AP information for accurate localization. Among the evaluated cases, the loss of AP3 results in the most notable performance degradation, followed by AP1, which is attributed to their corner placement and proximity to obstacles (M2 and M3), leading to stronger attenuation and multipath effects.

The visual comparisons in Figures 10-13 further illustrate the distribution of ADE in the training iterations and the spatial error distribution in the workspace for each failure case. Regions with low localization error (blue-green) dominate across all settings, confirming robust performance in most areas. Higher errors (orange-red) primarily appear near room boundaries and corners, where signal reception weakens due to increased distance from active APs. The maximum error reaches approximately 3.5 m in extreme edge cases; however, the majority of positions remain below the error threshold of 1.0 m, demonstrating that the proposed framework maintains reliable positioning even when one AP is unavailable.

4.3 Comparison with Previous Works

Recent advancements in IPS have leveraged both fingerprint-based techniques and deep learning enhancements to improve localization accuracy. Previous studies that also utilize Wi-Fi technology for indoor positioning are reviewed below for comparative analysis. A summary of these works is presented in Table 4. In [36], the authors proposed TILoc, a robust fingerprinting method that utilizes

Table 3: Optimized network configurations and ADE after 200 iterations of the model with 7 blocks under different AP failure scenarios

Faulty AP	**Filters**	**Dropout**	**Learning Rate**	**Network Parameters**	**ADE**
AP1	(128, 64, 32, 32, 128, 512, 128)	0.1	0.0001	1,328,130	0.776
AP2	(32, 256, 256, 512, 512, 512, 128)	0.3	0.0001	7,164,322	0.842
AP3	(32, 256, 256, 64, 64, 256, 64)	0.3	0.0001	2,614,498	1.019
AP4	(128, 512, 128, 32, 256, 32, 256)	0.2	0.0001	1,450,050	0.858
4 APs are fine	(64, 128, 32, 256, 64, 32, 128)	0.1	0.0001	393,410	0.552

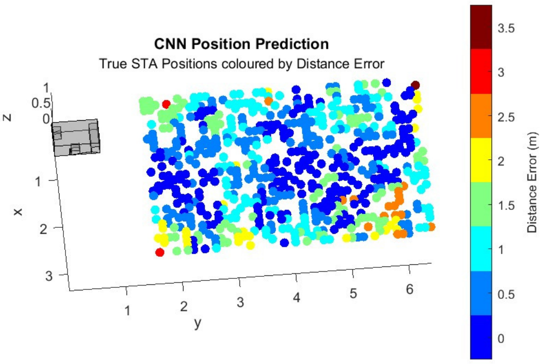

Figure 10: Mobile robot position prediction with the optimal positions of four APs and with AP1 failure

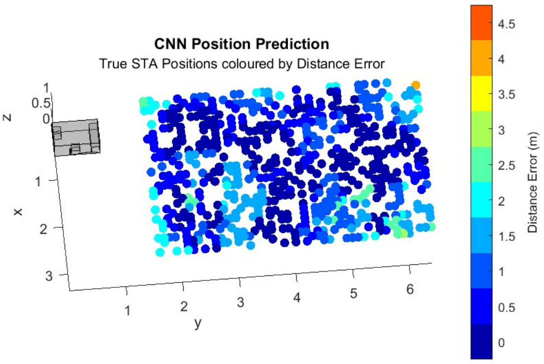

Figure 11: Mobile robot position prediction with the optimal positions of four APs and with AP2 failure

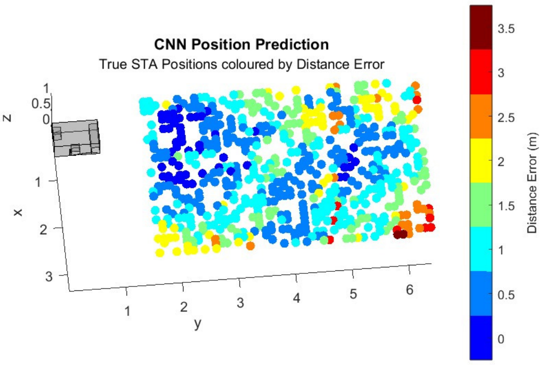

Figure 12: Mobile robot position prediction with the optimal positions of four APs and with AP3 failure

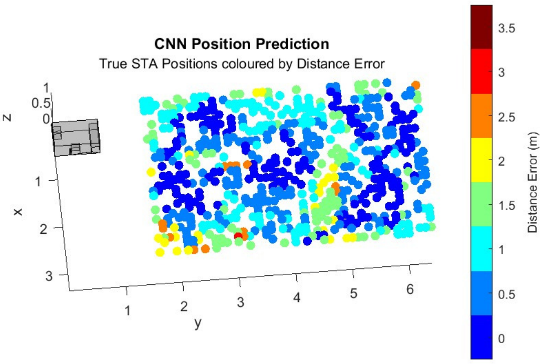

Figure 13: Mobile robot position prediction with the optimal positions of four APs and with AP4 failure

a multi-fingerprint (MF) database to accommodate temporal variations in RSSI values. By integrating

Table 4: Comparison of the proposed method to previous works

Publication	Method base	Year of publication	ADE (m)
[36]	KNN	2020	2.5
[37]	DL	2021	1.6
[38]	DL	2022	1.73
[8]	DL	2023	0.93
[32]	LSTM	2024	1.99
[39]	Bayesian optimization	2025	1.03
This work	DL	-	0.55

a resilient matching algorithm alongside outlier detection, TILoc achieved a mean positioning error of 2.5 m, surpassing conventional KNN and Weighted KNN (WKNN) methods. In [37], a hybrid framework was introduced that combines a Stacked Improved Sparse Autoencoder (SISAE) with a RNN. The SISAE is responsible for extracting high-level features from RSS data, while the RNN captures temporal dependencies, resulting in a mean positioning error of 1.60 m, particularly effective in dynamic environments. Study [38] conducted a comparative analysis of several deep learning models, including multilayer perceptrons, one-dimensional and two-dimensional convolutional neural networks (1D/2D CNNs), and LSTM for RSS-based localization. Among these, the second variant of the LSTM model demonstrated superior performance, achieving a root mean square error (RMSE) of 1.73 m in real-time tracking scenarios. In [8], the authors proposed a DNN architecture augmented with a state-based positioning mechanism that incorporates information from previous states to enhance real-time performance. This approach eliminates the requirement for signal averaging and attained a mean positioning error of 0.93 m on both benchmark and custom datasets. Reference [32] presented a Bayesian optimization-based LSTM model, which optimized hyperparameters for enhanced accuracy and employed dimensionality reduction techniques such as Truncated Singular Value Decomposition (TSVD) and Linear Discriminant Analysis (LDA). The model achieved a mean positioning error of 1.99 m on a publicly available dataset. In [39], a dual clustering strategy combined with Bayesian probabilistic optimization was introduced, which segments the environment according to spatial and signal domain characteristics. This method provided a mean positioning error of 1.03 m, outperforming conventional clustering-based fingerprinting techniques. For comparison, in this work, the proposed approach achieved a mean distance error of 0.55 m, indicating a substantial improvement in the localization accuracy relative to existing solutions.

5 Conclusion

This paper presents a comprehensive framework to improve the accuracy, efficiency, and resilience of IPS through optimized AP deployment and DNN architectures. By integrating Channel Impulse Responses with a DCNN, the proposed method effectively minimizes the average distance error while also reducing the number of APs required. The use of a Simulated Annealing algorithm facilitates optimal AP placement under varying constraints, balancing accuracy and deployment cost. Additionally, we employ the Hyperband algorithm for efficient hyperparameter tuning, improving both model performance and computational efficiency across multiple DNN architectures. To address real-world deployment challenges, such as AP failures, we introduce a fault-tolerant mechanism that preserves system robustness without significant performance degradation. Experimental results demonstrate that the proposed approach outperforms conventional fingerprinting methods, especially in complex and dynamic indoor environments. Furthermore, the flexibility of the framework allows it to adapt to various application domains, including factory automation, mining, and warehouse operations. By jointly optimizing the number of APs, the placement, and the configuration of the model, our solution contributes to a more scalable and cost-effective IPS deployment strategy.

Although the effectiveness of the proposed framework has been demonstrated through rigorous ray-tracing simulations, we recognize that experimental validation in real-world settings is an essential next step. Despite the use of realistic electromagnetic propagation models and complex multipath accounting, practical deployments may encounter additional variables. These include temporal signal fluctuations, hardware calibration inaccuracies, human mobility, and environmental dynamics that are often difficult to replicate fully in a simulated environment. Consequently, future work will explore real-time deployment scenarios, dynamic AP reconfiguration, and the integration of additional sensing modalities to further enhance localization accuracy and robustness. In general, this research offers valuable insights into the development of next-generation IPS solutions that meet the growing demands of intelligent indoor environments.

Acknowledgment

This research is funded by Vietnam National Foundation for Science and Technology Development (NAFOSTED) under grant number 102.02-2023.06. The authors would like to thank Dr. Shaik Mohammed Salman of Hitachi Energy AB for his valuable discussions and insights.

Conflict of interest

The authors declare no conflict of interest.

References

[1] Zafari, F.; Gkelias, A.; Leung, K.K. (2019). A Survey of Indoor Localization Systems and Technologies, *IEEE Communications Surveys & Tutorials*, 21(3), 2568–2599, 2019.

[2] Nessa, A.; Adhikari, B.; Hussain, F.; Fernando, X.N. (2020). A Survey of Machine Learning for Indoor Positioning, *IEEE Access*, 8, 214945–214965, 2020.

[3] Boysen, N.; De Koster, R.; Weidinger, F. (2019). Warehousing in the e-commerce era: A survey, *European Journal of Operational Research*, 277(2), 396–411, 2019.

[4] Vanhie-Van Gerwen, J.; Geebelen, K.; Wan, J.; Joseph, W.; Hoebeke, J.; De Poorter, E. (2021). Indoor drone positioning: Accuracy and cost trade-off for sensor fusion, *IEEE Transactions on Vehicular Technology*, 71(1), 961–974, 2021.

[5] He, S.; Chan, S.-H.G. (2016). Wi-Fi Fingerprint-Based Indoor Positioning: Recent Advances and Comparisons, *IEEE Communications Surveys & Tutorials*, 18(1), 466–490, 2016.

[6] Shang, S.; Wang, L. (2022). Overview of WiFi fingerprinting-based indoor positioning, *Iet Communications*, 16(7), 725–733, 2022.

[7] Brunato, M.; Battiti, R. (2005). Statistical learning theory for location fingerprinting in wireless LANs, *Computer Networks*, 47(6), 825–845, 2005.

[8] Nabati, M.; Ghorashi, S.A. (2023). A real-time fingerprint-based indoor positioning using deep learning and preceding states, *Expert Systems with Applications*, 213, 118889, 2023.

[9] Dao, V.-L.; Salman, S.M. (2022). Deep neural network for indoor positioning based on channel impulse response, In *2022 IEEE 27th International Conference on Emerging Technologies and Factory Automation (ETFA)*, 1–8, 2022.

[10] Li, L.; Jamieson, K.; DeSalvo, G.; Rostamizadeh, A.; Talwalkar, A. (2018). Hyperband: A novel bandit-based approach to hyperparameter optimization, *Journal of Machine Learning Research*, 18(185), 1–52, 2018.

[11] Namee, K.; Kaewkajone, T.; Thierchot, T.; Meny, A.; Kaewsaeng-On, R.; Nimyungdee, C. (2024). Improving Indoor Navigation Accuracy with Neural Networks: A Focus on Signal Propagation Challenges, In *Proceedings of the 2024 13th International Conference on Networks, Communication and Computing*, 32–37, 2024.

[12] Chapre, Y.; Mohapatra, P.; Jha, S.; Seneviratne, A. (2013). Received signal strength indicator and its analysis in a typical WLAN system (short paper), In *38th Annual IEEE Conference on Local Computer Networks*, 304–307, 2013.

[13] Yang, Z.; Zhou, Z.; Liu, Y. (2013). From RSSI to CSI: Indoor localization via channel response, *ACM Computing Surveys (CSUR)*, 46(2), 1–32, 2013.

[14] Zhao, W.; Han, S.; Hu, R.Q.; Meng, W.; Jia, Z. (2018). Crowdsourcing and multisource fusion-based fingerprint sensing in smartphone localization, *IEEE Sensors Journal*, 18(8), 3236–3247, 2018.

[15] Kjærgaard, M.B.; Munk, C.V. (2008). Hyperbolic Location Fingerprinting: A Calibration-Free Solution for Handling Differences in Signal Strength (concise contribution), In *2008 Sixth Annual IEEE International Conference on Pervasive Computing and Communications (PerCom)*, 110–116, 2008.

[16] Guo, X.; et al. (2018). Knowledge aided adaptive localization via global fusion profile, *IEEE Internet Things Journal*, 5(2), 1081–1089, 2018.

[17] Yu, K.; Wen, K.; Li, Y.; Zhang, S.; Zhang, K. (2019). A Novel NLOS Mitigation Algorithm for UWB Localization in Harsh Indoor Environments, *IEEE Transactions on Vehicular Technology*, 68(1), 686–699, 2019.

[18] Poulose, A.; Han, D.S. (2020). UWB indoor localization using deep learning LSTM networks, *Applied Sciences*, 10(18), 6290, 2020.

[19] Han, S.; Li, Y.; Meng, W.; Li, C.; Liu, T.; Zhang, Y. (2019). Indoor Localization With a Single Wi-Fi Access Point Based on OFDM-MIMO, *IEEE Systems Journal*, 13(1), 964–972, 2019.

[20] AlHajri, M.I.; Ali, N.T.; Shubair, R.M. (2019). Indoor Localization for IoT Using Adaptive Feature Selection: A Cascaded Machine Learning Approach, *IEEE Antennas and Wireless Propagation Letters*, 18(11), 2306–2310, 2019.

[21] Maranò, S.; Gifford, W.M.; Wymeersch, H.; Win, M.Z. (2010). NLOS identification and mitigation for localization based on UWB experimental data, *IEEE Journal on Selected Areas in Communications*, 28(7), 1026–1035, 2010.

[22] Rezgui, Y.; Pei, L.; Chen, X.; Wen, F.; Han, C. (2017). An efficient normalized rank based SVM for room level indoor WiFi localization with diverse devices, *Mobile Information Systems*, 2017, 6268797, 2017.

[23] Wu, Z.; Fu, K.; Jedari, E.; Shuvra, S.R.; Rashidzadeh, R.; Saif, M. (2016). A Fast and Resource Efficient Method for Indoor Positioning Using Received Signal Strength, *IEEE Transactions on Vehicular Technology*, 65(12), 9747–9758, 2016.

[24] Wymeersch, H.; Marano, S.; Gifford, W.M.; Win, M.Z. (2012). A Machine Learning Approach to Ranging Error Mitigation for UWB Localization, *IEEE Transactions on Communications*, 60(6), 1719–1728, 2012.

[25] Zhu, H.; Cheng, L.; Li, X.; Yuan, H. (2023). Neural-network-based localization method for Wi-Fi fingerprint indoor localization, *Sensors*, 23(15), 6992, 2023.

[26] Hoang, M.T.; Yuen, B.; Dong, X.; Lu, T.; Westendorp, R.; Reddy, K. (2019). Recurrent Neural Networks for Accurate RSSI Indoor Localization, *IEEE Internet of Things Journal*, 6(6), 10639–10651, 2019.

[27] Cai, M.; Lin, Z. (2023). Precise WiFi Indoor Positioning using Deep Learning Algorithms, *arXiv preprint arXiv:2307.02011*, 2023.

[28] Wang, X.; Wang, X.; Mao, S. (2020). Deep Convolutional Neural Networks for Indoor Localization with CSI Images, *IEEE Transactions on Network Science and Engineering*, 7(1), 316–327, 2020.

[29] Le, D.V.; Meratnia, N.; Havinga, P.J.M. (2018). 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)*, 1–7, 2018.

[30] Liu, K.; Zhang, H.; Ng, J.K.-Y.; Xia, Y.; Feng, L.; Lee, V.C.S.; Son, S.H. (2018). Toward Low-Overhead Fingerprint-Based Indoor Localization via Transfer Learning: Design, Implementation, and Evaluation, *IEEE Transactions on Industrial Informatics*, 14(3), 898–908, 2018.

[31] Vishwakarma, R.; Joshi, R.B.; Mishra, S. (2023). IndoorGNN: A graph neural network based approach for indoor localization using WiFi RSSI, In *International Conference on Big Data Analytics*, Springer, 150–165, 2023.

[32] Nguyen, D.K.; Nguyen, L.C.; Hoang, M.K. (2024). A Bayesian Optimization Based Deep Learning Model for Wi-Fi Fingerprinting based Indoor Positioning, *Journal of Communications*, 19(10), 2024.

[33] Yun, Z.; Iskander, M.F. (2015). Ray Tracing for Radio Propagation Modeling: Principles and Applications, *IEEE Access*, 3, 1089–1100, 2015.

[34] Liu, J.; Wang, X. (2016). Physical layer authentication enhancement using two-dimensional channel quantization, *IEEE Transactions on Wireless Communications*, 15(6), 4171–4182, 2016.

[35] IEEE (2023). IEEE Draft Standard for Information technology–Telecommunications... (802.11be), *IEEE P802.11be/D3.0*, 1–999, 2023.

[36] Li, H.; Qian, Z.; Tian, C.; Wang, X. (2020). TILoc: Improving the robustness and accuracy for fingerprint-based indoor localization, *IEEE Internet of Things Journal*, 7(4), 3053–3066, 2020.

[37] Bai, J.; Sun, Y.; Meng, W.; Li, C. (2021). Wi-Fi Fingerprint-Based Indoor Mobile User Localization Using Deep Learning, *Wireless Communications and Mobile Computing*, 2021, 6660990, 2021.

[38] Karakusak, M.Z.; Kivrak, H.; Ates, H.F.; Ozdemir, M.K. (2022). RSS-based wireless LAN indoor localization and tracking using deep architectures, *Big Data and Cognitive Computing*, 6(3), 84, 2022.

[39] Chen, M.; Pu, Q. (2025). Bayesian optimized indoor positioning algorithm based on dual clustering, *Scientific Reports*, 15(1), 9272, 2025.

Cite this paper as:

Van-Phuc Hoang; Thi-Kieu-Lan Ta; Van-Lan Dao; Thi-Yen Hoang. "High-Accuracy Indoor Positioning Using Deep Neural Networks Combining with Optimal Hyperparameters and Fault-Tolerant Mechanism", *International Journal of Computers Communications & Control*, 21(2), 7398, 2026.
https://doi.org/10.15837/ijccc.2026.2.7398