LCVAE-CNN: Indoor Wi-Fi fingerprinting CNN positioning method based on LCVAE

Shixun Wu, Xinrui Zeng, Miao Zhang, *Member, IEEE,* Kanapathippillai Cumanan, *Senior Member, IEEE,* Abdulhamed Waraiet, Zheng Chu, *Member, IEEE,* and Kai Xu

***Abstract*—While Wi-Fi Received Signal Strength Indicator (RSSI) fingerprinting has emerged as a prominent solution for indoor positioning, its accuracy remains hindered by labor-intensive data collection and environmental variability. To overcome these challenges, we propose a novel LCVAE-CNN methodology that integrates a Location-Conditioned Variational Autoencoder (LCVAE) and a multi-task Convolutional Neural Network (CNN) to enhance data quality and positioning performance. The LCVAE employs a dual-encoder architecture to augment RSSI fingerprints by jointly modeling signal features and spatial dependencies, introducing three key innovations: (1) dual-stream encoding that decouples RSSI and location processing for more effective feature learning, (2) a geospatial loss function that enforces topological consistency in the generated data, and (3) conditional data augmentation that preserves physical constraints of indoor spaces. The multi-task CNN then leverages shared feature extraction to jointly optimize classification and regression tasks, enabling efficient and accurate positioning. Extensive evaluations on the UJIIndoorLoc and Tampere datasets demonstrate the superiority of the LCVAE-CNN that achieves 98.80% floor classification accuracy with a Mean Positioning Error (MPE) of 6.79 meters on UJIIndoorLoc, whereas 97.22% accuracy with a MPE of 5.44 meters on the Tampere dataset. Compared to five state-of-the-art methods, it improves floor accuracy by at least 1.9% and reduces MPE by over 19%, while maintaining comparable computational overhead, thereby achieving superior accuracy-efficiency tradeoffs.**

***Index Terms*—Indoor positioning; LCVAE; CNN; Data augmentation**

I. Introduction

LOCATION-BASED services (LBS) have changed the way people live, playing an increasingly pivotal role in navigation, logistics, outdoor gaming, critical personnel tracking, disaster relief, and advertising etc. To facilitate the acquisition of location information of people and objects, researchers have developed various positioning schemes to support LBS [1]. Outdoor positioning primarily relies on satellite technologies like Global Positioning System (GPS) to determine users' geographic locations, while indoor positioning employs technologies such as WiFi [2][3], Bluetooth [4], Radio Frequency Identification (RFID) [5], and Ultra-Wideband (UWB) [6] to provide precise user location information, addressing the issue of weak and unstable GPS signals indoors. Among these, WiFi-based indoor positioning is particularly popular, owing to the simplicity of WiFi system deployment. No additional or specialized infrastructure required and easy implementation on various WiFi-enabled mobile devices [7].

WiFi-based indoor positioning methods are commonly categorized into two primary approaches: geometric positioning, which estimates location based on geometric relationships such as trilateration or triangulation using signal metrics like time of arrival (ToA) or angle of arrival (AoA), and fingerprinting positioning, which relies on prior signal measurements at known locations to infer position through pattern matching [8]. Although geometric positioning theoretically offers high accuracy, it faces several practical challenges such as dependence on high-precision hardware, environmental factors like multipath effects, complex deployment and maintenance, often resulting in escalated costs and constrained practicality [9]. Fingerprinting positioning methods leverage existing wireless network infrastructure, significantly mitigating technical and economic barriers via straightforward data collection and analysis processes, rendering them particularly apt for resource-limited or rapid deployment environments [10]. Fingerprinting-based positioning techniques can leverage a wide range of physical-layer features, including magnetic field variations, acoustic signals, and visible light intensity, depending on the sensing modality and application scenario. Generally, fingerprinting techniques in WiFi systems employ received signal strength indication (RSSI) or Channel State Information (CSI) to predict target location [11][12]. Notably, CSI can capture subtle channel changes including signal attenuation, phase, and reflection, handle multipath effects and signal interference, thus enhancing the stability and precision of CSI-based fingerprinting positioning systems [13]. However, CSI necessitates specific hardware, such as advanced network interface cards, which are not typically integrated into consumer devices, thereby limiting its widespread application [14]. In contrast, RSSI-based fingerprinting methods do not need extra hardware and can use existing wireless devices

S. Wu, X. Zeng, M. Zhang, and K. Xu are with the School of Information Science and Engineering, Chongqing Jiaotong University, Chongqing, 400074, China (email: wushixun333@163.com, 635338716@qq.com, msczz@foxmail.com, xkxjxwx@hotmail.com).

K. Cumanan and A. Waraiet are with the School of Physics, Engineering and Technology, University of York, York, UK (email: kanapathippillai.cumanan@york.ac.uk, abdulhamed.waraiet@york.ac.uk).

Z. Chu is with the Department of Electrical and Electronic Engineering, University of Nottingham Ningbo China, Ningbo 315100, China. (Email: andrew.chuzheng7@gmail.com)

This research was supported in part by the Natural Science Foundation of Chongqing under Grant CSTB2024NSCQ-MSX0275. The work of A. Waraiet and K. Cumanan was supported by the UK Engineering and Physical Sciences Research Council (EPSRC) under Grant EP/X01309X/1. The work of Zheng Chu was supported in part by the Ningbo Natural Science Foundation under Grant 2024J233.

for positioning, significantly reducing deployment costs and technical barriers.

The RSSI fingerprinting positioning approach typically comprises two phases: an offline phase and an online phase. During the offline phase, RSSI data is collected at multiple reference points (RPs) within an indoor environment to construct a "fingerprint" database, and then train the positioning model. This phase is critical for RSSI fingerprinting positioning, and it has several key challenges. First, the data collection process is labor-intensive, necessitating extensive RSSI data gathering at numerous indoor locations. Second, environmental changes, such as new obstacles or modifications to existing structures, can impact RSSI data, thereby diminishing the accuracy and utility of the database. Moreover, hardware variations between devices may lead to data inconsistencies, thereby increasing system complexity and instability [15]. To address these challenges, researchers are progressively exploring the transition from traditional machine learning (ML) [16] to deep learning (DL) [17] to achieve high-precision, low-cost positioning in complex, multi-building, and multi-floor environments. Although ML-based positioning models excel at handling complex scenarios and adapting to device variability, they predominantly rely on abundant labeled data for offline training. The substantial time and effort required for data acquisition may not be sustainable in practical applications [18]. Consequently, to overcome limitations of data acquisition and enhance training efficiency and model generalizability, generative models such as Generative Adversarial Networks (GANs) [19] and Variational Autoencoders (VAEs) [20] have been integrated into fingerprinting positioning systems. These models learn from limited labeled data and generate new, previously unseen fingerprint data, thereby expanding the fingerprint database and enhancing the adaptability and robustness of model to new environments [21].

In this paper, we integrate conditional variables into the VAE architecture, directly embedding geographic location information into the generative process of the model for data augmentation. This approach not only produces labeled positioning data for practical application, but also significantly reduces the complexity of subsequent data labeling, while ensuring high data quality and model flexibility. Further, this new fingerprint database is used to train a multi-task Convolutional Neural Network (CNN) positioning model. The dual-output structure of multi-task learning enhances the system efficiency and performance. The main contributions are as follows:

- We propose a data augmentation method based on the Location-Conditioned Variational Autoencoder (LCVAE), featuring a dual-encoder structure and a geographic information loss function with a weighting mechanism. This method enhances the geographic accuracy and practicality of the generated positioning data. Further, data matching and fingerprint selection algorithms are utilized to address sample imbalances and eliminate the impact of outliers, significantly improving floor classification precision and positioning accuracy.
- We design a DL-based multi-task CNN model to handle the high-dimensional feature space of RSSI data. This model can capture spatial and frequency features through two convolutional layers, enabling precise floor classification and accurate location prediction about user device, respectively.
- Combining LCVAE with multi-task CNN models, we propose a LCVAE-CNN fingerprint positioning method. This method has been extensively tested and validated on two public indoor positioning datasets, UJIIndoorLoc [22] and Tampere [23]. It is demonstrated that our proposed positioning method has superior performance in various indoor environments, highlighting its potential and effectiveness in complex indoor environment.

The rest of this paper is organized as follows: Section II reviews related work of WiFi-based positioning field. Section III introduces the proposed LCVAE-CNN fingerprint positioning method. Experimental results and analyses are discussed in Section IV. Finally, conclusions and future work are presented in Section V.

II. RELATED WORK

A. Traditional ML Approaches

The field of indoor positioning technologies is undergoing rapid advancements, enabling a wide array of applications such as indoor navigation and augmented reality. Among various techniques, fingerprinting-based approaches have become particularly prevalent due to their simplicity and ability to leverage existing infrastructure [24]. Early studies employed traditional ML algorithms such as K-Nearest Neighbor (KNN) and Support Vector Machines (SVM) for WiFi fingerprinting positioning [25]. For example, Hoang et al. [26] proposed a Soft Range-Limited KNN (SRLKNN) algorithm, and Zhang et al. [27] enhanced SVM models by incorporating user posture, thereby reducing localization errors. Recently, Huang et al. [28] proposed WiLoc, a long-term WiFi localization method using a lightweight Siamese Neural Network and KNN for final position estimation, effectively mitigating the degradation of localization accuracy over time. However, these methods often exhibit suboptimal performance in large-scale and dynamic environments due to inherent limitations in feature extraction capabilities.

B. DL-Based Fingerprinting Methods

To overcome the limitations of ML, recent research has turned to DL, which offers superior feature extraction capabilities and adaptability. Innovative approaches such as Recurrent Neural Networks (RNNs) [29], autoencoders [30], and hybrid models integrating CNNs with stacked autoencoders [31] have demonstrated notable performance gains in complex indoor environments. For instance, CCPos [32] employs a Convolutional Denoising Autoencoder (CDAE) to learn invariant and noise-resistant representations, improving the robustness of the model against signal fluctuations. In another study, Alitaleshi et al. [33] introduced Extreme Learning Machine Autoencoders (ELM-AE) for efficient feature mapping and dimensionality reduction, followed by a CNN for final location estimation, achieving enhanced positioning accuracy. More recently, Zhang et al. [34] introduced WiDAGCN, a Domain Adversarial Graph Convolutional Network designed to enhance

RSSI-based indoor positioning by leveraging graph structures and crowdsensed data. However, its performance heavily relies on the quality of graph construction, requiring precise node and edge definitions, along with careful optimization and hyperparameter tuning. These dependencies, coupled with the computational cost of graph processing, pose significant challenges for real-time and large-scale deployment, particularly in dynamic environments. Ayinla et al. [35] proposed a WiFi indoor localization method that integrates Recursive Feature Elimination with Cross-Validation (RFECV) and Deep Neural Networks with Batch Normalization (DNNBN). While the approach demonstrates high accuracy in building and floor classification, it still struggles with computational complexity and the need for extensive training iterations. Furthermore, Yu et al. [36] proposed WiNCDN, a WiFi-assisted non-cooperative indoor localization system using a dropout-based neural network, which efficiently tracks targets in both line-of-sight (LoS) and non-line-of-sight (NLoS) scenarios, offering improved accuracy and robustness compared to existing methods.

C. Fingerprint Enhancement Techniques

Although recent DL methods have demonstrated remarkable advancements in indoor positioning, their real-world deployment frequently encounters practical limitations. Specifically, they require large-scale labeled datasets, which are costly and labor-intensive to collect. Moreover, environmental dynamics and device heterogeneity frequently introduce data sparsity and inconsistency [37]. To address these challenges, fingerprint enhancement has emerged as a critical technique to improve data availability and model robustness. Initial efforts focused on interpolation-based methods [38] and signal reconstruction using autoencoders [39]. More recently, Yu et al. [40] introduced Combinatorial Data Augmentation (CDA) to bridge geometry-driven and data-driven WiFi positioning methods, demonstrating its effectiveness through experiments with WiFi RTT and IMU data. While CDA improves positioning accuracy, its reliance on integrating both methods may require further optimization for large-scale deployment in dynamic environments. In addition, generative models have also been explored as a means of enhancing fingerprint data, offering a powerful approach to further improve dataset diversity and model performance. Njima et al. [41] proposed a GAN-based approach that generated synthetic RSSI data from a small set of labeled samples, leveraging semi-supervised learning to assign pseudo-labels. Similarly, Qian et al. [42] introduced a semi-supervised VAE to enrich fingerprint databases and enhance generalization under sparse conditions. While these methods improve performance in data-scarce scenarios, they often struggle to capture fine-grained spatial variations in complex indoor environments. Among recent generative approaches, Wang et al. [43] proposed DeepRSSI, a CVAE enhanced with a sequential gate self-attention mechanism, to generate high-quality virtual RSSI fingerprint maps. This model effectively captures both spatial and temporal dependencies in RSSI data, thereby reducing reliance on manual site surveys and improving localization performance across various positioning algorithms. However, its effectiveness depends heavily on the similarity between training and deployment environments, which may limit its generalizability in structurally heterogeneous settings.

Notably, generative models still face challenges in reconstruction fidelity and feature disentanglement. Standard VAE often suffer from blurry reconstructions and limited precision, which can hinder location-specific learning. Semi-supervised VAEs may also entangle label values with latent features, reducing their representational power. To mitigate these issues, conditional VAE have been introduced, using floor labels as conditioning variables [32]. However, differences in data scale and sparsity between RSSI and coordinate inputs can lead to unbalanced learning and degraded performance. To overcome these limitations, we propose a LCVAE combined with a CNN-based positioning model in this paper. The LCVAE adopts a dual-encoder structure and introduces a geographic information loss to ensure spatial consistency and balance between input modalities. This framework enhances data generation quality and boosts positioning performance, addressing key shortcomings of existing VAE and CVAE models.

To better illustrate the distinction between our proposed method and existing approaches, we present a comparative summary in Table I. This table highlights key differences in model architectures, the implementation of data augmentation, and the specific limitations each method addresses.

III. Methodology

The LCVAE-CNN fingerprint positioning method, depicted in Figure 1, comprises both offline and online phases. In the offline phase, data from the original fingerprint database undergoes preprocessing and standardization to meet the requirements of subsequent model training. The LCVAE model enhances the preprocessed data by generating new fingerprint data for specified floors, thereby improving model generalization through increased data diversity. Subsequently, data matching and fingerprint selection further optimize the fingerprint database. Finally, the enhanced fingerprint data is used to train a multi-task CNN positioning model. In the online phase, real-time WiFi fingerprint data is input into the pretrained model, which outputs floor and location information, thereby providing users with immediate LBS.

A. Data Preprocessing

Data preprocessing is a fundamental step in the initial stages of data analysis and ML, employing various techniques for cleaning, scaling, and encoding data [44]. To generate fingerprints with location coordinate labels for designated floors, this process begins with data cleaning, removing irrelevant attributes such as user IDs and space IDs, focusing on significant features for positioning, such as longitude, latitude, floor, and building IDs. Additionally, to meet model requirements, floor and building IDs are combined into a new feature ‘floor_building’. For example, ‘Building ID1, Floor 0’ becomes ‘01’. Then, one-hot encoding is performed. Subsequently, RSSI values and location coordinates including longitude and latitude are normalized and scaled linearly

TABLE I
COMPARISON OF REPRESENTATIVE INDOOR POSITIONING METHODS

Method	Model Architecture	Data Augmentation	Limitations Addressed
WiDeep [30]	Stacked Denoising Autoencoder + Probabilistic Framework	✗	Handles RSSI noise but lacks data expansion mechanisms
CNNLoc [31]	Stacked Autoencoder + 1D CNN	✗	Limited generalization across buildings/floors; no data synthesis
CCPos [32]	Convolutional Denoising Autoencoder (CDAE)	✗	Learns robust features; no augmentation or balancing
EA-CNN [33]	Extreme Learning Machine Autoencoder + CNN	✓	Improves accuracy but lacks scalability and data diversity handling
WiDAGCN [34]	Domain-Adversarial Graph Convolutional Network	✗	Sensitive to graph construction; unstable training and high computational cost
GAN-based [41]	GAN + Semi-supervised Learning	✓	Generates synthetic data but lacks spatial structure control
VAE-based [42]	Semi-supervised Variational Autoencoder	✓	Improves generalization but suffers from blurry and entangled latent features
DeepRSSI [43]	CVAE + Self-attention + Temporal Modeling	✓	Captures spatiotemporal patterns; limited generalizability to new environments
Ours	**LCVAE + Multi-task CNN + Geo-loss**	✓	**Addresses data sparsity, sample imbalance, and spatial accuracy via dual-encoder learning and geospatial loss function**

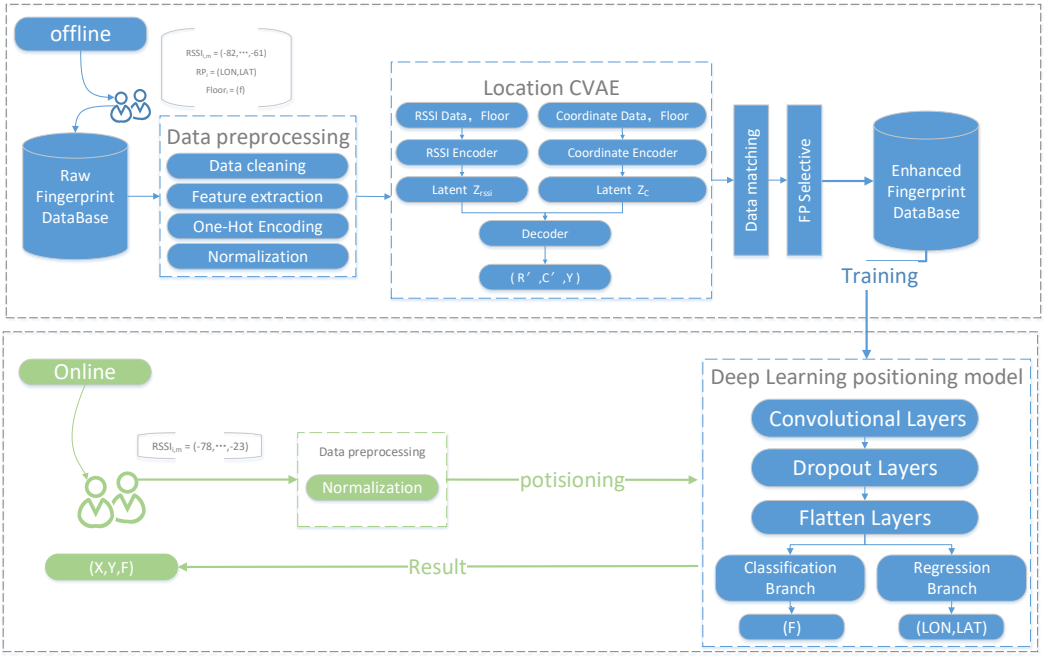


Fig. 1. Structure of LCVAE-CNN fingerprint positioning method.

between 0 and 1 to ensure equal influence within the model. The specific normalization formula is as follows:

$$x' = \frac{x - \min(x)}{\max(x) - \min(x)}, \tag{1}$$

where x represents the original data, and x' denotes the normalized data. To ensure data consistency, parameters used for the transformation are stored after normalization. This enables the recovery of the original data during the reverse normalization process.

B. LCVAE Data Augmentation

In this study, we propose a LCVAE model for data augmentation, extending the standard CVAE architecture, solving the problem of data scarcity and imbalance in indoor positioning. The structure of LCVAE, depicted in Figure 2, includes encoding and decoding components.

1) LCVAE Encoding: The LCVAE encoding architecture includes two distinct encoders, one is used for RSSI data and the other is designed for geographic coordinate data. Each encoder extracts and encodes the characteristics of the input data through a multi-layer fully connected network that includes batch normalization, *LeakyReLU* activation functions,

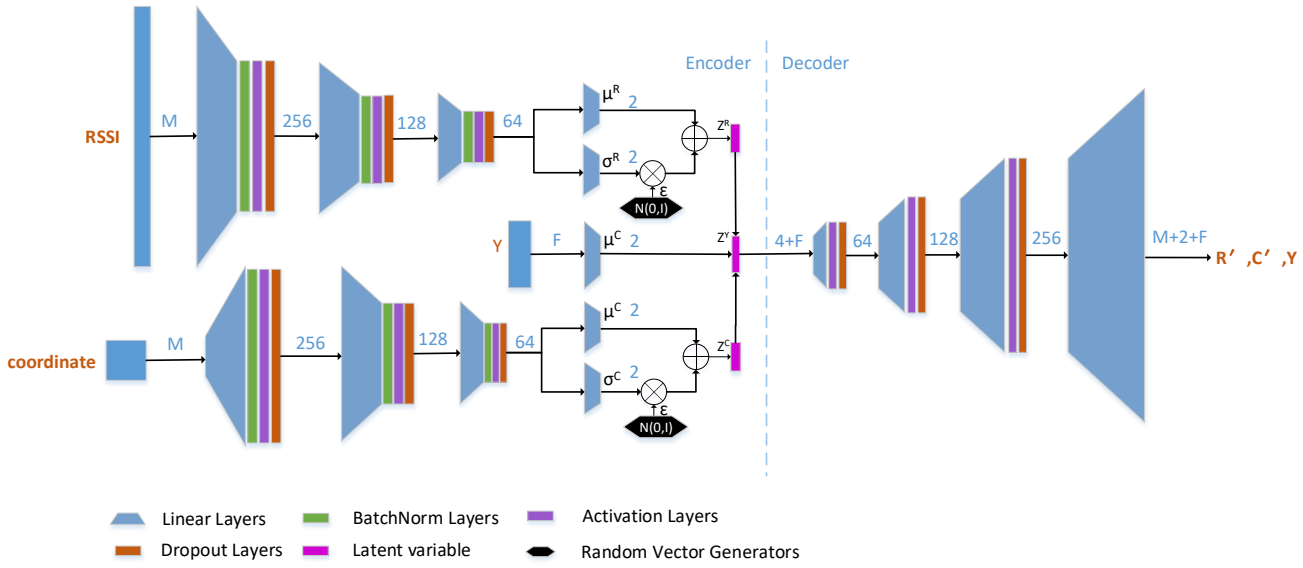


Fig. 2. Structure diagram of LCVAE.

and Dropout layers. RSSI data is represented as an $N \times M$ matrix, where N is the number of RPs, or samples, and M denotes the number of WiFi access points. The expression of it is given as follows:

$$R = \begin{bmatrix} R_{1,1} & R_{1,2} & \cdots & R_{1,M} \\ R_{2,1} & R_{2,2} & \cdots & R_{2,M} \\ \vdots & \vdots & \ddots & \vdots \\ R_{N,1} & R_{N,2} & \cdots & R_{N,M} \end{bmatrix}, \tag{2}$$

where $R_{i,j}$ indicates the i-th RSSI measurement sample from the j-th WiFi access point.

Geographic coordinate data $C = (C_1, \ldots, C_N)$, where $C_i = (\text{LON}_i, \text{LAT}_i)$ represent the i-th RP location, and $Y \in \mathbb{R}^F$ contains the floor and building conditional information, where F denotes the total number of floor-building combinations and varies depending on the dataset. The encoding function maps the input data $X = (R, C)$ to the distribution of the latent space variable Z. The posterior distribution of Z, $q(Z \mid X)$, is typically assumed to be Gaussian [45], approximating a standard normal distribution $q(Z)$. This distribution $q(Z \mid X)$ is derived from X through a neural network:

$$\log q(Z \mid X) = \log \mathcal{N}(Z; \mu(X), \sigma^2(X)), \tag{3}$$

where $\mu(X)$ and $\sigma^2(X)$ are the functions parameterized by the neural network.

We define two encoders E_R and E_C to learn feature representations from R and C, respectively. Subsequently, these representations are utilized within the LCVAE model. The outputs of encoders E_R and E_C are expressed as:

$$(\mu_{R,i}, \log \sigma^2_{R,i}) = E_R(R), \tag{4}$$

$$(\mu_{C,i}, \log \sigma^2_{C,i}) = E_C(R). \tag{5}$$

By independently processing RSSI and geographic coordinate data, we can more meticulously explore the internal structure and correlations of each data type without worrying about noise or information loss that might arise from mixing these two feature types in the early stages. Subsequently, the features derived from the RSSI and geographic coordinate encoders are used to estimate their respective means and logarithmic variances. These parameters can define the Gaussian distributions of the latent space variables Z_R and Z_C, which are sampled through reparameterization from these distributions:

$$Z_{R,i} = \mu_{R,i} + \epsilon \cdot \exp\left(-\frac{1}{2} \log \sigma^2_{R,i}\right), \tag{6}$$

$$Z_{C,i} = \mu_{C,i} + \epsilon \cdot \exp\left(-\frac{1}{2} \log \sigma^2_{C,i}\right), \tag{7}$$

where ϵ represents noise sampled from a standard normal distribution.

Except for the input data X, each layer of the encoder and decoder also receives additional conditional information Y [46]. This information is extracted directly from the annotated labels in the UJIIndoorLoc and Tampere datasets, which contain explicit floor and building identifiers for each reference point. The conditional variable is one-hot encoded to ensure compatibility with neural network processing. In the LCVAE model, each conditional input is processed through a linear layer to compute specific latent space means $\mu_{y,i}$, while the variance is kept constant. This design allows the model to adjust the means of the latent space variables according to different types of inputs without altering the variance of distribution. By incorporating Y into both the encoder and decoder of the LCVAE, the model is guided to generate samples under specific location contexts, thereby improving the spatial consistency of augmented data and ensuring that

synthesized fingerprints remain relevant to their intended environment. Once the latent space variables $Z_{R,i}$ and $Z_{C,i}$ are obtained, they are concatenated with the input condition Y to produce the encoder output $Z_{Y,i}$. Without such conditional information, the model may generate semantically inconsistent or physically implausible data, especially in multi-floor or multi-building scenarios.

2) LCVAE Decoding: The task of LCVAE decoding is to map the latent space variables $Z_{Y,i}$ output from the encoder back to the original data space X, ensuring that the generated simulated samples $X' = (R', C')$ closely resemble the original input X. This can be expressed as:

$$X' \sim p(X \mid Z; \theta), \tag{8}$$

where θ represents the parameters of the decoder, characterizing the specific computation method of the decoding process. This process involves a probabilistic generative model that allows sampling from a probability distribution to obtain new samples that are close to the true data distribution. During the training stage, the optimization of θ is achieved by minimizing the loss function. This not only aids the model in accurately learning the underlying structure of the data, but also ensures that the generated data maintains statistical consistency with the original data. The decoder model includes a fully connected network that reconstructs the original input data from the combined latent space variables and conditional information. In addition, the decoder network progressively enlarges the feature dimensions, ultimately outputting data in the original dimensions.

3) Loss Function: To ensure the quality of data generated by the LCVAE model, this paper combines reconstruction loss and Kullback-Leibler (KL) divergence loss, and proposes a new loss function L, defined as follows:

$$L = \alpha \cdot L_{recon}(X', X) + \beta \cdot L_{KL}(X, Y) + L_{geo}(C', C), \tag{9}$$

where L_{recon} represents the reconstruction loss, a metric that quantifies the discrepancy between the model decoder output X' and the original input data X. In the LCVAE model, the decoder reconstructs input data based on the latent space variables $Z_{Y,i}$ and the conditional variable Y. The reconstruction loss ensures that the decoder can effectively use the latent space variables to reproduce the original input data, thereby learning useful data features and structures. The formula is as follows:

$$L_{recon} = E_{q(Z|X)}[\log p(X \mid Z; \theta)]. \tag{10}$$

In practical applications, it is challenging to obtain an analytical expression of L_{recon}. Given that the squared error directly impacts the performance of model in specific tasks such as prediction, classification, or generation, it is generally substituted for (10) with the following specific formula:

$$L_{recon} = \frac{1}{N}\sum_{i=1}^{N} \|\mathbf{R}_i - \mathbf{R}'_i\|^2 + \frac{1}{N}\sum_{i=1}^{N} \sqrt{\|\mathbf{C}_i - \mathbf{C}'_i\|^2}, \tag{11}$$

where $\|.\|$ represents the Euclidean norm. $\mathbf{R}_i$ and $\mathbf{R}'_i$ are the RSSI vectors for the original and reconstructed data of the i-th sample, respectively. Similarly, $\mathbf{C}_i$ and $\mathbf{C}'_i$ are the coordinate vectors for the original and reconstructed data of the i-th sample, respectively.

L_{KL} represents the KL divergence loss, which is used to quantify the difference between the posterior distribution of the latent space variable Z, $q(Z \mid X)$, and its prior distribution $q(Z)$. The formula of this loss function is as follows:

$$KL(q(Z \mid X) \| q(Z)) = -\frac{1}{2}\sum_{i} \left(1 + \log(\sigma_i^2) - (\mu_i)^2 - (\sigma_i)^2\right). \tag{12}$$

KL divergence is introduced for two main reasons: one is to ensure regularization of the latent space variable distribution, preventing overfitting; the other is to push the posterior distribution closer to the prior distribution, enhancing the generalization capability of the model. This loss function helps the model explore more effective latent representations during learning, increasing the diversity of generated outputs. Given that the LCVAE model requires a unique mean for each class of samples, the specific formula for KL divergence in this model is:

$$L_{KL} = L_{KL,R} + L_{KL,C}, \tag{13}$$

$$L_{KL,R} = -\frac{1}{2}\sum_{k} \left(1 + \log(\sigma_{R,i}^2) - (\mu_{R,i} - \mu_{R,j})^2 - (\sigma_{R,i}^2)\right),$$

$$L_{KL,C} = -\frac{1}{2}\sum_{i} \left(1 + \log(\sigma_{C,i}^2) - (\mu_{C,i} - \mu_{C,j})^2 - (\sigma_{C,i}^2)\right),$$

where α and β are the weighting factors for the reconstruction and KL divergence loss, respectively.

To enhance the accuracy of generated RP coordinates, an additional geographic information loss function L_{geo} is introduced. This loss function aims to ensure that the generated coordinate data C' closely approximates the true coordinates C and is reasonable in spatial distribution, making the data more geographically meaningful and suitable for positioning tasks. The mathematical expression is as follows:

$$L_{geo}(C', C) = \gamma \cdot \frac{1}{n}\sum_{i=1}^{n} \|C'_i - C_i\| + \lambda \cdot \text{Penalty}(C', C), \tag{14}$$

where $\|C'_i - C_i\|$ is the Euclidean distance between the generated and true coordinates for the i-th sample. γ adjusts the relative importance of the base loss, and λ controls the impact of the centroid offset penalty.

To ensure that generated coordinates are not only accurate for individual samples but also collectively approximate the center of true coordinates, a centroid offset penalty is introduced. This penalty addresses the deviations of generated coordinates from the center of true coordinates, primarily used for generating data for individual buildings. Its expression is as follows:

$$\text{Penalty}(C', C) = \frac{1}{N}\sum_{i=1}^{N} \left\| C'_i - \frac{1}{N}\sum_{i=1}^{N} C_i \right\|. \tag{15}$$

The purpose of this design is to balance the accuracy of individual samples and the clustering effect of generated samples, thereby enhancing the accuracy and practicality of the whole prediction. The loss function of LCVAE can help generate models to not only reconstruct data points, but also

learn the distribution characteristics of the data, making the generated data with geographic information more reliable and practical. In practical applications, indoor environments may have complex building structures and obstacles that affect signal propagation. However, the geographic information loss helps the model learn how to maintain high accuracy in positioning, and guides the model to prioritize learning to reduce the geographic location error. This assists the model in adjusting parameters more effectively during learning, enhancing learning efficiency.

It is important to note that the fingerprint data sourced from the UJIIndoorLoc and Tampere datasets were collected in realistic indoor environments that naturally include various obstacles, such as walls, furniture, and human activity. These environmental factors influence signal propagation and result in signal attenuation, reflection, and multipath effects, all of which are inherently captured in the RSSI measurements. Notably, the proposed LCVAE model does not assume an obstacle-free environment. Instead, by conditioning on location-related variables, i.e., floor and building identifiers, and incorporating a geographic consistency loss, our proposed model can learn the statistical patterns associated with different spatial regions, including the signal distortions caused by obstacles. Consequently, the model is capable of synthesizing fingerprint data that reflect the complex propagation characteristics observed in real-world indoor environments. Even when trained on data with obstacles, the model can produce plausible samples under similar conditions, thus maintaining fidelity to the underlying spatial structures.

C. Data Matching

For multi-level or multi-building datasets, the accuracy of classification models decreases with more generated fingerprints [47]. To ensure that the data input into the positioning model is evenly distributed and accurate across all floors, we propose a data matching algorithm. This algorithm sets the model to evaluation mode, preventing the changed weights of model during data augmentation. For each specified floor label Y, it first identifies the corresponding data indices from the floor list F, then randomly selects a designated number of samples S. Thus, the number of the total generated data is $S \times F$. This random selection allowing for repetitions ensures that sample quantity requirement is met even when available data is low. The selected samples undergo forward propagation through the model to generate new data. This process does not involve backpropagation, therefore does not affect model parameters.

In the phase of data matching, the algorithm processes each selected sample individually. For each sample, its RSSI measurements and coordinate data are fed into the model to generate new data points, while the floor label remains unchanged. The generated samples are combined with the original floor labels of sample to form a complete data record, resulting in the generated dataset D_{gen}, thereby providing more balanced and accurate training dataset for the positioning model. Algorithm 1 summarizes the aforementioned process.

Algorithm 1 Data Matching Algorithm

Require: Model (LCVAE), R (RSSI data), C (coordinate data), Y (floor labels), F (list of floors), S (number of samples per floor)
Ensure: D_{gen} (generated dataset)

```
Set the model to be evaluation mode
Initialize the generated dataset D_gen
for each Y in F do
    Locate the current floor's index in Y, termed F_index
    if data exists for the current floor then
        Randomly select S indices from F_index, termed selected_indices
        for each index in selected_indices do
            Extract the corresponding R and C
            Generate new data points (R', C') using LCVAE model
            Add the predicted output (R', C') as augmented data points to D_gen
        end for
    end if
end for
return the generated dataset D_gen
```

D. Fingerprint Selection

Selecting an appropriate distance between real and generated fingerprints is crucial to reduce noise and interference. This study employs a KD-tree for spatial proximity filtering of both real and generated fingerprint data points to enhance the quality of dataset. The core of this method uses Euclidean distance as the metric to assess the spatial closeness between generated and original data points, resulting in a refined dataset D_{sel}. Initially, the original dataset $D_{ori} = (R, C, Y)$ and the generated dataset $D_{gen} = (R', C', Y)$ are inputted. A KD-tree is constructed using the geographic coordinates $C_i = (LON_i, LAT_i)$ from D_{ori}, enabling fast nearest neighbor queries. A distance threshold r is then set to filter generated data points $C'_i = (LON'_i, LAT'_i)$ that are close to the original data points. Each data point in the generated dataset undergoes a nearest neighbor query, returning indices of original data points within the threshold. By iterating over the query results, generated data points that meet the criteria are selected, and corresponding data points are extracted from D_{gen} to form D_{sel}. Subsequently, D_{sel} and D_{ori} are then merged to create the final enhanced fingerprint database. The specific Algorithm 2 is as follows.

E. Multi-task CNN Model

In this study, we develop a DL-based multi-task CNN model tailored for indoor WiFi positioning, jointly addressing tasks of floor classification and coordinate regression, as shown in Fig. 3. Unlike prior works that typically train separate models for classification and regression, our approach leverages a unified architecture with shared convolutional layers. This enables the model to learn generalized low-level features that benefit both tasks, fostering mutual reinforcement while reducing redundancy. While multi-task learning has been widely studied

Algorithm 2 Fingerprint Selection Algorithm

Require: D_{ori} (original dataset), D_{gen} (generated dataset), r (distance threshold)
Ensure: D_{sel} (selected dataset)
Construct a KD-tree using geographic coordinates from D_{ori}
Set proximity threshold $r \leftarrow 5$
Query D_{gen} for data points at or below distance threshold r
$Indices \leftarrow$ tree_query_ball_point$(D_{gen}.values, r)$, initialize selection list as empty
for each i, idx enumerate $Indices$ **do**
  if $idx \neq 0$ **then**
    Selected_indices.append(idx)
  end if
end for
$D_{sel} \leftarrow D_{gen}.iloc[Selected\_indices]$
return selected dataset D_{sel}

in other domains such as computer vision and time-series analysis, it remains underexplored in WiFi fingerprint-based indoor localization, where prior works typically train separate models or use task-specific branches without explicit joint optimization.

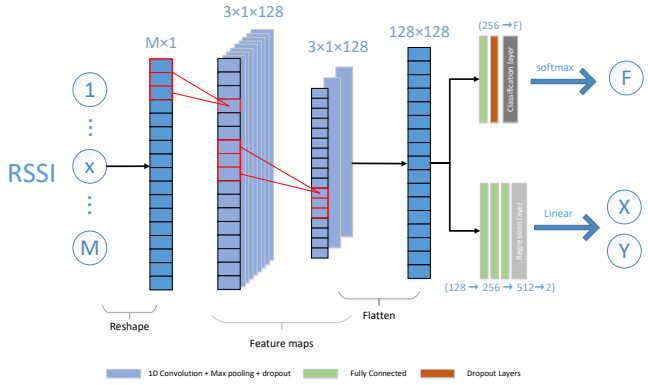


Fig. 3. Structure diagram of multi-task CNN positioning model.

To empirically validate the efficiency of this design, we compare the complexity of our multi-task CNN with that of training two independent CNN models: classification and regression. The standalone classification model contains approximately 4.25 million trainable parameters and incurs 34.26 MFLOPs, while the regression model comprises 2.31 million parameters and 30.39 MFLOPs. In contrast, our unified model requires only 3.33 million parameters and 13.24 MFLOPs, reducing the parameter count by 49.2% and computation cost by 79.5%, as shown in Table II. These results confirm that our unified architecture not only simplifies the training and deployment pipeline, but also improves memory efficiency and testing speed, making it more suitable for real-time applications and resource-constrained environments.

The input consists of a normalized RSSI vector with dimensions $M \times 1$, which is sequentially processed by two convolutional blocks. Each block includes a one-dimensional convolutional layer with a kernel size of 3×1 and 128 filters, followed by a max pooling layer and a dropout layer. This design enables the network to extract robust features from the input signal.

TABLE II
COMPARISON OF MODEL COMPLEXITY

Model	Parameters	FLOPs	Architecture
Classification-only	4.25M	34.26M	Separate CNN
Regression-only	2.31M	30.39M	Separate CNN
Ours (Joint)	**3.33M**	**13.24M**	Unified Multi-task CNN

The extracted feature maps (128×128) are flattened and fed into two task-specific branches: a classification branch and a regression branch. The classification branch predicts the specific floor level of the device using fully connected layers and a softmax activation function. Conversely, the regression branch, also composed of fully connected layers, employs a linear activation function to output precise location coordinates.

The model is optimized using Adam with a moderate learning rate.This ensures effective handling of the complex and high-dimensional nature of RSSI data while maintaining computational efficiency. To counteract overfitting, dropout layers are incorporated in the classification branch but omitted in the regression branch to prevent loss of crucial feature information, which is essential for absolute positioning tasks. Early stopping is implemented to halt training and prevent further loss reduction. A combination of classification and regression losses is employed to simultaneously optimize these two tasks. The performance of the multi-task CNN model is assessed based on the loss and accuracy on a validation set, ensuring its generalization capability on unseen data and effectiveness in handling complex indoor positioning, especially dealing with high-dimensional RSSI data. The detailed hyperparameter settings of the proposed multi-task CNN are summarized in Table III.

TABLE III
MULTI-TASK CNN POSITIONING MODEL PARAMETERS.

Parameter	Value
Epoch	50
Batch Size	64
Early Stopping Patience	10
Max Pooling	2
CNN Activation Function	ReLU
Regression Activation Function	Linear
Regression Loss	MSE
Regression Metric	MAE
Classification Activation Function	SoftMax
Classification Loss	Categorical Cross-Entropy
Classification Metric	Floor Accuracy
Learning Rate	0.0001
L2 Regularization Factor	0.01
Optimizer	Adam
Filter Size	3
Number of Filters	128
Dropout Factor	0.5

F. Training and Testing Algorithms

To improve clarity and reproducibility, the training and testing procedures of the proposed LCVAE-CNN framework are summarized in Algorithm 3 and Algorithm 4.

The proposed LCVAE-CNN framework consists of a dual-encoder LCVAE and a multi-task CNN. Specifically, the LCVAE module contains approximately 395,466 trainable parameters, and the CNN module includes 3,327,695 parameters, resulting in a combined total of about 3.72 million trainable parameters. We further conducted a computational complexity analysis using TensorFlow Profiler for the CNN component, which revealed that a single forward pass requires approximately 13.24 million FLOPs. The majority of computation is concentrated in the convolutional and dense layers, with the 'Conv1D' and 'Dense' layers accounting for over 95% of the total operations. These results are in line with the findings in Table II, which compare our unified multi-task CNN with standalone models for classification and regression. This demonstrates that the proposed framework is suitable for real-time indoor positioning applications, even in resource-constrained environments such as edge computing devices.

Algorithm 3 Training Procedure of LCVAE-CNN Model

Require: RSSI samples $\mathbf{R}$, coordinates $\mathbf{C}$, floor labels $\mathbf{Y}$
Ensure: Trained LCVAE and CNN models
Initialize parameters of LCVAE and CNN
for each epoch **do**
  for each mini-batch $(\mathbf{R}_b, \mathbf{C}_b, \mathbf{Y}_b)$ **do**
    // Stage 1: LCVAE Data Augmentation
    Encode $\mathbf{R}_b$, $\mathbf{C}_b$, $\mathbf{Y}_b$ into latent variables
    Sample latent vectors and decode to generate $\hat{\mathbf{R}}, \hat{\mathbf{C}}, \hat{\mathbf{Y}}$
    Compute LCVAE loss and update parameters
    // Stage 2: CNN Multi-task Learning
    Train CNN with original + generated samples
    Compute multi-task loss and update parameters
  end for
end for
return Trained models

Algorithm 4 Testing Procedure of LCVAE-CNN

Require: Trained CNN model, testing RSSI input $\mathbf{R}_{\text{test}}$
Ensure: Predicted floor $\hat{\mathbf{Y}}$, coordinates $\hat{\mathbf{C}}$
Normalize input $\mathbf{R}_{\text{test}}$
Feed into CNN for forward pass
Output floor label via softmax, $\hat{\mathbf{Y}}$
Output coordinates via linear layer, $\hat{\mathbf{C}}$
return $\hat{\mathbf{Y}}, \hat{\mathbf{C}}$

Furthermore, we clarify the nature of the input and generated data used during training by emphasizing that the inputs to the LCVAE are real-world RSSI vectors and their corresponding geographic coordinates, collected from indoor environments with complex structural characteristics. In contrast, the generated samples are not direct duplicates but are synthesized through a stochastic decoding process from latent representations conditioned on the floor-building context. From a data augmentation perspective, the LCVAE framework generates new samples by sampling latent vectors from Gaussian distributions learned during training. These are modulated by both the extracted features and the conditional input $\mathbf{Y}$, enabling the decoder to produce signal data and coordinates that are spatially consistent. Additionally, the geographic consistency loss enforces structural plausibility in the generated coordinates while avoiding overlap with the original samples.

This generative process ensures diversity in both signal and spatial domains, allowing the augmented dataset to cover a broader and smoother distribution. Importantly, the generated samples do not cause overfitting; instead, they enrich the training dataset with variations that improve model robustness. As demonstrated in our experiments, models trained with LCVAE-augmented data consistently outperform those trained on the original dataset alone, validating the effectiveness of our conditional generation strategy.

IV. Experimental Results and Analysis

All experiments were conducted on a workstation with an AMD Ryzen 7 7840HS CPU and 32 GB RAM, running Windows 11. The implementation was developed using PyTorch and TensorFlow in a Python 3.9 environment within PyCharm.

A. Dataset Description

We utilize two public indoor positioning datasets UJIIndoorLoc [22] and Tampere [23] for experimental validation and analysis. The UJIIndoorLoc dataset, developed by Jaume I University in Valencia, Spain, is specifically designed for the development and evaluation of indoor positioning systems. It includes 21,049 Wi-Fi fingerprint samples collected from 933 RPs across a range of devices and users, where 19,938 samples are divided into training sets and the remaining 1,111 are used as validation sets, covering multiple floors within multiple buildings. Each data point contains RSSI data from 520 WiFi access points, as well as geographic location information including building ID (0, 1, 2), floor level (0-4), and coordinates. This dataset has been widely adopted for evaluating and comparing various machine learning-based indoor localization methods. The Tampere dataset is collected through a crowdsourcing approach in a five-story university building at the University of Tampere in Finland, comprising a total of 4,648 fingerprints from 21 different devices, where 697 samples are randomly designated as training data and the remaining 3,951 ones are as validation data. Each fingerprint includes RSSI data from up to 992 WiFi access points and corresponding geographic location information, and represents a unique RP, as no repeated measurements were performed during crowdsourced data collection. These two datasets are randomly and evenly divided into 20% data from the training set as the test set, and undetected WiFi access points in each fingerprint are defaulted to +100dbm. The features of these two datasets are shown in Table IV. For brevity, some column headers in the table use standard abbreviations: Blds = number of buildings, Flrs = number of floors, Samples = number of training/validation samples.

Assess the positioning accuracy of the model, Mean Positioning Error (MPE) is introduced as the metric. This metric

TABLE IV
MAIN FEATURES OF THE UJIINDOORLOC AND TAMPERE DATASETS.

Dataset	Blds	Flrs	RPs	Samples	APs
UJI	3	4/5	933	19,938 / 1,111	520
Tampere	1	5	697 / 3951	697 / 3,951	992

is calculated by determining the Euclidean distance between the true geographic coordinates and the predicted geographic coordinates, as described below:

$$MPE = \frac{1}{N}\sum_{i=1}^{N}\sqrt{(\mathrm{LON}_i' - \mathrm{LON}_i)^2 + (\mathrm{LAT}_i' - \mathrm{LAT}_i)^2}, \quad (16)$$

where $(\mathrm{LON}_i, \mathrm{LAT}_i)$ and $(\mathrm{LON}_i', \mathrm{LAT}_i')$ represent the true and predicted geographic coordinates of the i-th test sample, respectively.

B. Model Parameter Optimization

1) Parameter Optimization of multi-task CNN model: In the process of parameters optimization about the multi-task CNN model, we firstly focus on the size of the convolution kernels and the number of filters, because these parameters directly impact the ability of model to capture data features. Various combinations of kernel sizes (3, 5, 7) and filter counts (64, 96, 128, 256, 512) are tested to determine the optimal configuration. As shown in Figure 4, a kernel size of 3 and 128 filters yields the best MPE performance on both the UJIIndoorLoc and Tampere datasets.

Dropout operations can enhance the robustness of the model by randomly dropping connections of the network during training, which are crucial for preventing overfitting. Therefore, further tuning of the dropout factor is necessary. By testing different values of the Dropout factor from 0.2 to 0.7, as shown in Figure 5, the floor classification accuracy and MPE of two datasets achieve relatively good results when the Dropout factor is 0.5.

This series of optimization steps have significantly enhanced the performance of our proposed model across various datasets, and the final parameters setting of the multi-task CNN model are detailedly shown in Table III.

2) Parameter Optimization of LCVAE Model: The LCVAE model includes an encoder dedicated to encoding the RSSI and coordinates, a reparameterization mechanism to represent the latent space, and a decoder for reconstructing the input. Therefore, the choice of optimizer is crucial for determining the efficiency and stability of model training. Comparisons of RMSProp and Adam optimizers for data augmentation as well as the original data are illustrated in Figure 6, experimental results indicate that RMSProp consistently outperforms Adam, regardless of the amount of training data. Furthermore, data augmentation with the LCVAE model always achieves higher positioning accuracy compared to the original data.

The weights of loss function α, β, γ and λ significantly influence the performance of LCVAE model. For reconstruction loss, we usually need a relatively large weight to ensure that the model can effectively learn the basic structure of the data, thereby setting $\alpha = 1$. KL divergence loss is used to maintain good statistical characteristics of latent representations. The

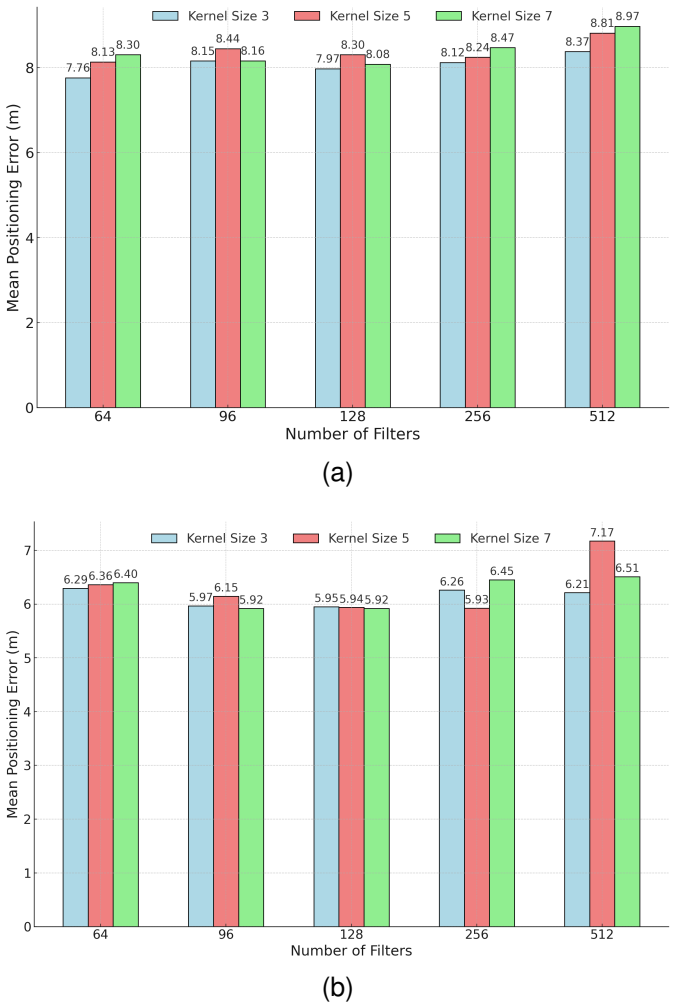


Fig. 4. MPE with different filter number and kernel size.(a) UJIIndoorLoc (b) Tampere.

weight is usually low to prevent it from overwhelming the reconstruction loss, resulting in a decline in the quality of the output generated by the model. Thus, we set $\beta = 0.1$. Finally, geographic information loss is a location-specific function to ensure that the outputted geographic coordinates of the model are close to the true coordinates. The selection of weights can be adjusted according to the need for positioning accuracy. However, if the weight is too large, it may cause the model to over-optimizing geographical data at the expense of other important features. Thus, a moderate weight $\gamma = 0.5$ is set. Since there are three buildings in the UJIIndoorLoc dataset, there is no need to punish them to move closer to the center. Thus, λ is set to 0. In addition, there is only one building in the Tampere dataset, and λ is set to 0.01. The comparison between the generated data and the original data is shown in Figure 7. It can be seen that the generated RPs maintain the geographical distribution of the original data, making them geographically significant and suitable for positioning tasks.

To further validate the effectiveness of the proposed geographic information loss function, the weight of the geographic loss function is set to $\gamma = 0$ in the Tampere dataset, as illustrated in Fig. 8(a). The generated data points do not

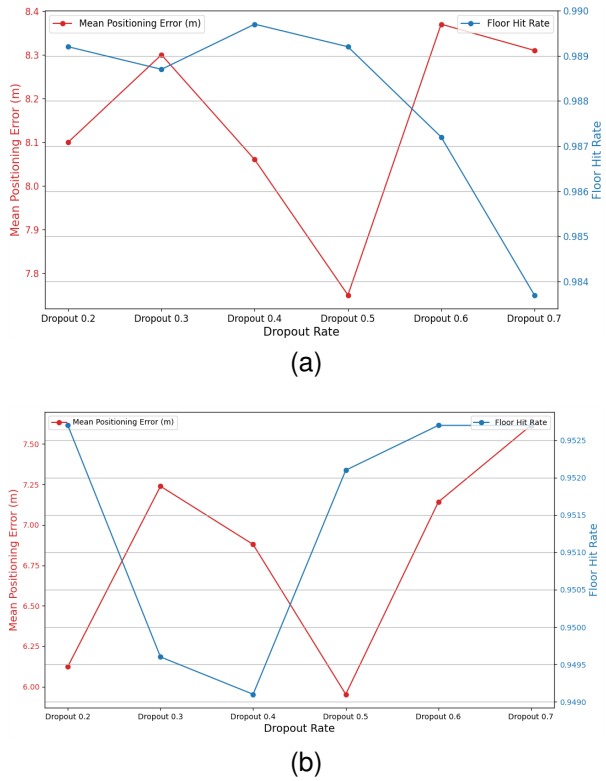


Fig. 5. Performance comparison for different Dropout Rate. (a) UJIIndoorLoc (b) Tampere.

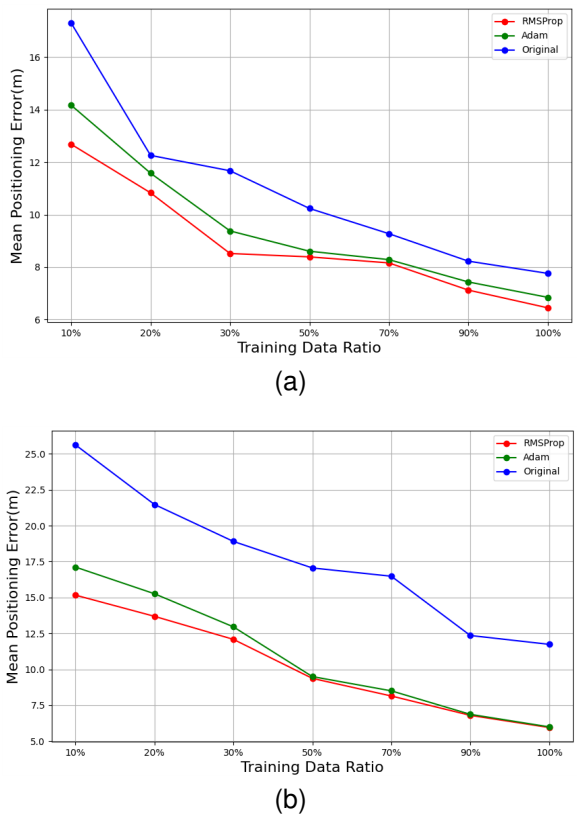


Fig. 6. Compare different optimizers. (a) UJIIndoorLoc (b) Tampere.

maintain the original geographic distribution. However, when the weight of the geographic loss function is set to $\gamma = 0.5$, as shown in Fig. 8(b), the generated data points are similar yet distinct from the original. This indicates that the geographic information loss function can help the model to generate data with meaningful geographic properties.

Through continuous experimentation and comparative analysis, the primary parameters of the LCVAE model are determined as shown in Table V.

TABLE V
LCVAE MODEL PARAMETERS.

Parameter	Value
Input Dimension	Number of WiFi Access Points
Coordinate Dimension	2
Hidden Layer Dimension	2
Conditional Dimension	Number of Floors
Optimizer	RMSprop
Learning Rate	0.001
Layers in RSSI Encoder	3 layers (256, 128, 64 nodes each)
Layers in Coordinate Encoder	3 layers (256, 128, 64 nodes each)
Encoder Activation Function	LeakyReLU
Dropout Factor	0.3
Decoder Structure	4 layers (64, 128, 256, WAPs+2 nodes)
Decoder Activation Function	Sigmoid
Loss Function	$L_{\text{recon}} + L_{KL} + L_{\text{geo}}$
Reconstruction Loss Weight α	1.0
KL Divergence Loss Weight β	0.1
Geographic Loss Weight γ	0.5
Penalty Weight λ	0.0/0.01

To visually demonstrate the effectiveness of the LCVAE data augmentation method, we performed dimensionality reduction using the t-SNE method [48] on both the original fingerprint data and the synthetic fingerprint data generated by LCVAE. We conducted experiments on both the UJIIndoorLoc and Tampere datasets to assess the distribution characteristics of the augmented data and its similarity to the original data. A random sample of 1,000 samples was taken from both the original and augmented data, and the distribution of these samples in the low-dimensional space is shown in Figure 9.

In the UJIIndoorLoc dataset, the fingerprint data exhibits a relatively concentrated pattern, forming distinct clusters. The synthetic fingerprint data generated by the LCVAE closely aligns with the original data's distribution, almost entirely covering the original data's spatial layout, while also filling in sparse regions with new fingerprint points, thereby achieving a more uniform data distribution. Due to the stable environment in which the UJIIndoorLoc data was collected, and the minimal impact of multipath effects on the RSSI data, the synthetic data generated by LCVAE aligns well with the original distribution, exhibiting no significant deviations.

In contrast, the original fingerprint data in the Tampere dataset is more dispersed, with a noticeably more irregular distribution compared to the UJIIndoorLoc dataset. This variability is attributable to the crowdsourced nature of the Tampere dataset, where varying RSSI measurements across different user devices lead to lower data stability and higher fluctuations due to the dynamic indoor environment. While the LCVAE-generated augmented data still follows the distribution trend of the original data, some regions show greater concentration, and larger deviations are observed in others. This phenomenon

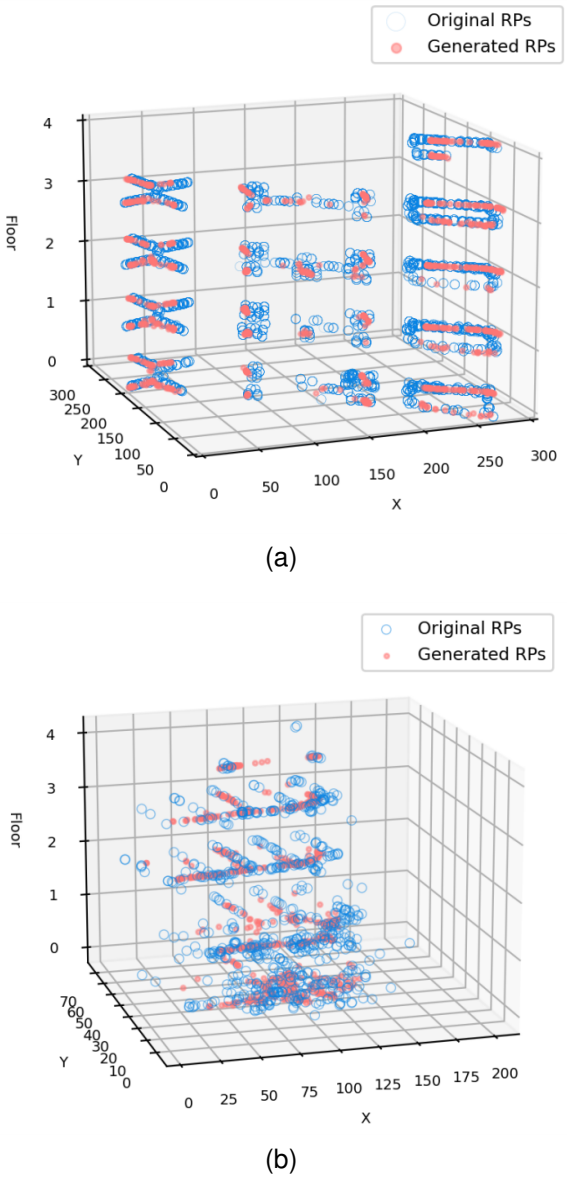


Fig. 7. Comparison of Raw and generated RPs distribution. (a) UJIIndoorLoc (b) Tampere.

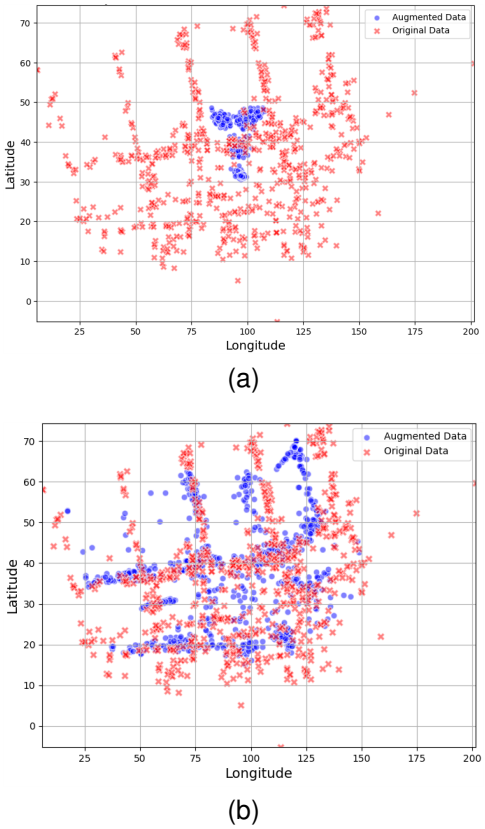


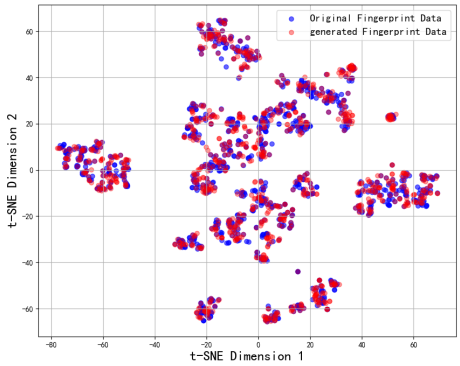
Fig. 8. Comparison of RPs distributionn. (a) $\gamma = 0$ (b) $\gamma = 0.5$.

can be attributed to the uneven distribution in the original data, where sparse or noisy areas cause the model to focus on the predominant distribution pattern, neglecting the less populated regions. Despite this, LCVAE effectively compensates for the sparse regions, enhancing the spatial distribution uniformity of the fingerprint data. This improvement in data distribution significantly enhances the model's adaptability to diverse environments and boosts positioning accuracy.

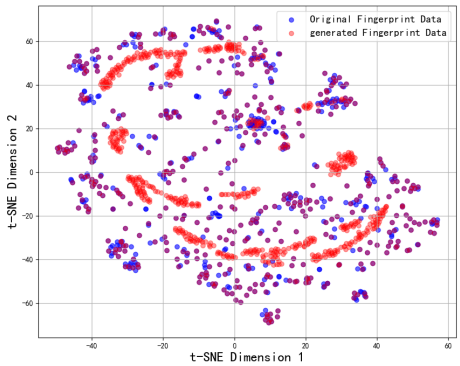
C. Positioning Results and Analysis

To visually demonstrate the effectiveness of floor classification and precisely reveal the performance of the classification model across different floors, the confusion matrices for floor classification using the proposed LCVAE-CNN fingerprint method on the UJIIndoorLoc and Tampere datasets are shown in Fig. 10. The confusion matrix in Fig. 10(a) indicates that the majority of floors are correctly classified, demonstrating that the model can effectively distinguish different floors and has a high discriminative ability. However, the confusion matrix in Fig. 10(b) reveals some classification errors between floors 0 and 1, suggesting that further optimization strategies may be required for classifying adjacent floors.

To comprehensively evaluate the performance of the proposed LCVAE-CNN fingerprint positioning method, it is compared and analyzed against five existing methods: SAE-CNN[31], CDAE-CNN[32], EA-CNN[33], HADNN[49], and CNNLoc[50]. Specifically, SAE-CNN adopts a stacked autoencoder to reduce the dimensionality of sparse RSSI data, followed by a CNN for position estimation, serving as a representative deep learning-based baseline. CDAE-CNN, utilized in the CCPos system, combines a convolutional denoising autoencoder with a CNN to enhance robustness against signal noise and to improve generalization performance. EA-CNN integrates an Extreme Learning Machine Autoencoder with a 2D-CNN for 3D indoor localization, making it effective in multi-floor environments. HADNN employs a hierarchical auxiliary deep neural network that incorporates building and floor-level labels into a regression framework to improve scalability in large indoor spaces. CNNLoc uses a stacked autoencoder and a 1D-CNN to provide multi-building and multi-floor classification, designed for complex indoor settings. For fair comparison and reproducibility, all baseline methods were re-implemented based on their original publications. The architectural structures, training configurations, and hyperparameters were followed as closely as possible. All methods were trained and evaluated under the same

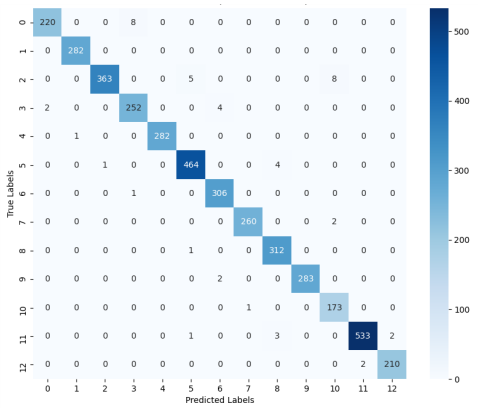


(a)

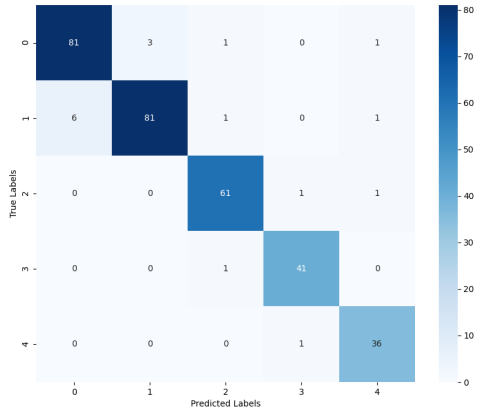


(b)

Fig. 9. Fingerprint data distribution comparison before and after data augmentation. (a) UJIIndoorLoc (b) Tampere



(a)



(b)

Fig. 10. Confusion matrix of floor hit. (a) UJIIndoorLoc (b) Tampere.

experimental settings, including input preprocessing, batch size, optimizer, and dataset partitioning, to ensure that the results are comparable and reproducible. The floor classification accuracy and MPE positioning performance on the UJIIndoorLoc and Tampere datasets are detailly presented in Table VI. For floor classification accuracy, the LCVAE-CNN method achieves 98.80% and 97.22% on the UJIIndoorLoc and Tampere datasets, respectively, showing improvements of 2.49% and 1.9% over the best existing methods. Moreover, compared with the multi-task CNN method without LCVAE, the accuracy of floor classification is improved to some extent, especially when the data volume is small. The MPE of LCVAE-CNN method is 6.79 meter and 5.44 meter on the UJIIndoorLoc and Tampere datasets, respectively. Compared to the highest positioning accuracy, the MPE is reduced by 19% and 32%, respectively. Additionally, the LCVAE data augmentation significantly reduces the MPE. Overall, the LCVAE-CNN method not only enhances floor classification precision but also significantly improves positioning accuracy.

Further comparison of the positioning accuracy is performed with the Cumulative Distribution Function (CDF). All methods are trained on the same dataset and evaluated on the same test set. In order to reduce the influence of preprocessing

TABLE VI
LCVAE-CNN POSITIONING PERFORMANCE COMPARED WITH STATE-OF-THE-ART METHODS.

Dataset	Model	Floor hit rate	MPE (m)
UJIIndoorLoc	HADNN[49]	93.15%	11.59
	CNNLoc[50]	95.19%	12.99
	SAE-CNN[31]	96.03%	10.78
	CDAE-CNN[32]	95.30%	12.4
	EA-CNN[33]	96.31%	8.34
	without LCVAE	98.65%	10.07
	LCVAE-CNN	**98.80%**	**6.79**
Tampere	HADNN[49]	94.58%	9.05
	CNNLoc[50]	95.32%	11.67
	SAE-CNN[31]	94.22%	10.8
	CDAE-CNN[32]	93.67%	10.83
	EA-CNN[33]	95.30%	7.96
	without LCVAE	96.46%	9.07
	LCVAE-CNN	**97.22%**	**5.44**

steps, simple sample transformation matrices are used for the CNN models. Fig. 11 illustrates the CDF performance of these methods on both datasets. As shown in Fig. 11(a), the proposed LCVAE-CNN method with blue curve outperforms other methods across all position error thresholds, demonstrating superior positioning accuracy. This improved performance is attributed to the application of LCVAE in the model, which significantly enhances the environmental perception and positioning capabilities of model through data augmentation. Fig. 11(b) highlights that the data augmentation of LCVAE

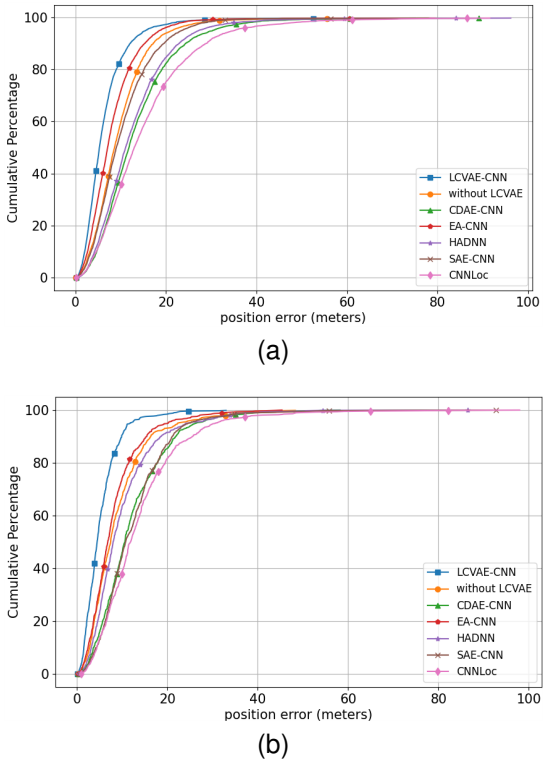


Fig. 11. CDF comparison of different positioning methods. (a) UJIIndoorLoc (b) Tampere.

is particularly pronounced in the Tampere dataset, showing a notable performance improvement than other methods. This indicates that our proposed method can effectively enhance positioning accuracy in the environment with sparse data.

To fully evaluate the overall performance of the LCVAE-CNN method, a comparison of the running time about different positioning methods is shown in Table V. The running time is obtained from testing 100 samples of each positioning method and repeating 100 times to calculate the average, ensuring the stability and reliability of the results. Although the running time of the LCVAE-CNN method is slightly higher than that of the method without LCVAE, it shows better running efficiency than other CNN-based positioning methods, such as CNNLoc, CDAE-CNN and EA-CNN. This demonstrates that the LCVAE-CNN method has high computational efficiency, while simultaneously maintaining high positioning accuracy.

TABLE VII
COMPARISON OF RUNNING TIME.

Positioning method	Running time (ms)
HADNN[49]	19.66
CNNLoc[50]	23.01
SAE-CNN[31]	22.68
CDAE-CNN[32]	25.01
EA-CNN[33]	23.92
without LCVAE	21.22
LCVAE-CNN	**22.53**

V. CONCLUSION AND FUTURE WORK

In this paper, a multi-task fingerprint positioning method based on the LCVAE-CNN was proposed. The dual encoder of LCVAE and the improved loss function enabled the generated data more geographically meaningful and more suitable for the positioning task. Our proposed method addressed the challenges in indoor positioning related to data collection and environmental variability, thereby significantly improving robustness of the positioning model. Compared to five existing fingerprint positioning methods, the experimental results showed that our proposed method achieves significant performance improvement on the two public datasets of UJIIndoorLoc and Tampere, especially in reducing positioning errors and improving the accuracy of floor classification.

Our future works will explore potential enhancements to the LCVAE architecture, including deeper networks, refined training algorithms, or more complicated loss functions to enhance the performance of model. Additionally, this research will be expanded to more diverse indoor environments, involving multi-story buildings and areas with challenging signal obstructions. Integrating various sensor data, such as inertial measurement units, ultrasonic, and optical sensors, may provide a more comprehensive understanding of the environment.

REFERENCES

[1] J. Dai, M. Wang, B. Wu, J. Shen, and X. Wang, "A survey of latest wi-fi assisted indoor positioning on different principles," *Sensors*, vol. 23, no. 18, p. 7961, 2023.

[2] X. Lin, J. Gan, C. Jiang *et al.*, "Wi-fi-based indoor localization and navigation: A robot-aided hybrid deep learning approach," *Sensors*, vol. 23, no. 14, p. 6320, 2023.

[3] S. Wu, W. Huang, M. Li and K. Xu, "A Novel RSSI Fingerprint Positioning Method Based on Virtual AP and Convolutional Neural Network," in *IEEE Sensors Journal*, vol. 22, no. 7, pp. 6898–6909, 1 April1, 2022.

[4] Y. Zhuang, J. Yang, Y. Li *et al.*, "Smartphone-based indoor localization with bluetooth low energy beacons," *Sensors*, vol. 16, no. 5, p. 596, 2016.

[5] S. Cheng, S. Wang, W. Guan *et al.*, "3dlra: An rfid 3d indoor localization method based on deep learning," *Sensors*, vol. 20, no. 9, p. 2731, 2020.

[6] A. Prorok and A. Martinoli, "Accurate indoor localization with ultra-wideband using spatial models and collaboration," *The International Journal of Robotics Research*, vol. 33, no. 4, pp. 547–568, 2014.

[7] C. Xu, B. Firner, Y. Zhang *et al.*, "The case for efficient and robust rf-based device-free localization," *IEEE Transactions on Mobile Computing*, vol. 15, no. 9, pp. 2362–2375, 2015.

[8] Y. Gu, A. Lo, and I. Niemegeers, "A survey of indoor positioning systems for wireless personal networks," *IEEE Communications Surveys & Tutorials*, vol. 11, no. 1, pp. 13–32, 2009.

[9] A. Owfi, C. C. Lin, L. Guo *et al.*, "A meta-learning based generalizable indoor localization model using channel state information," in *GLOBECOM 2023 - 2023 IEEE Global Communications Conference*. IEEE, 2023, pp. 4607–4612.

[10] T. Yang, A. Cabani, and H. Chafouk, "A survey of recent indoor localization scenarios and methodologies," *Sensors*, vol. 21, no. 23, p. 8086, 2021.

[11] T. K. Geok, K. Z. Aung, M. S. Aung *et al.*, "Review of indoor positioning: Radio wave technology," *Applied Sciences*, vol. 11, no. 1, p. 279, 2020.

[12] X. Liu, R. Wu, H. Zhang, Z. Chen, Y. Liu and T. Qiu, "Graph Temporal Convolutional Network-Based WiFi Indoor Localization Using Fine-Grained CSI Fingerprint," in *IEEE Sensors Journal*, vol. 25, no. 5, pp. 9019–9033, Mar.2025.

[13] Y. Zhang, C. Qu, and Y. Wang, "An indoor positioning method based on csi by using features optimization mechanism with lstm," *IEEE Sensors Journal*, vol. 20, no. 9, pp. 4868–4878, 2020.

[14] R. S. Naser, M. C. Lam, F. Qamar *et al.*, "Smartphone-based indoor localization systems: A systematic literature review," *Electronics*, vol. 12, no. 8, p. 1814, 2023.

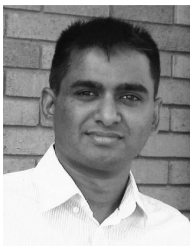
[15] X. Zhu, W. Qu, T. Qiu *et al.*, "Indoor intelligent fingerprint-based localization: Principles, approaches and challenges," *IEEE Communications Surveys & Tutorials*, vol. 22, no. 4, pp. 2634–2657, 2020.

[16] A. Chriki, H. Touati, and H. Snoussi, "Svm-based indoor localization in wireless sensor networks," in *Proc. 2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 1144–1149.
[17] N. Hernández, I. Parra, H. Corrales *et al.*, "Wifinet: Wifi-based indoor localisation using cnns," *Expert Systems with Applications*, vol. 177, p. 114906, 2021.
[18] C. Li and Y. Mao, "Improved indoor localization algorithm combining k-means clustering algorithm and wasserstein generative adversarial network algorithm," in *2023 19th International Conference on Natural Computation, Fuzzy Systems and Knowledge Discovery (ICNC-FSKD)*. IEEE, 2023, pp. 1–5.
[19] W. Njima, A. Bazzi, and M. Chafii, "Dnn-based indoor localization under limited dataset using gans and semi-supervised learning," *IEEE Access*, vol. 10, pp. 69 896–69 909, 2022.
[20] B. Chidlovskii and L. Antsfeld, "Semi-supervised variational autoencoder for wifi indoor localization," in *2019 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2019, pp. 1–8.
[21] S. A. Junoh and J. Y. Pyun, "Enhancing indoor localization with semi-crowdsourced fingerprinting and gan-based data augmentation," *IEEE Internet of Things Journal*, vol. 11, no. 7, pp. 11 945–11 959, 2024.
[22] A. M.-U. J. Torres-Sospedra, R. Montoliu *et al.*, "Ujiindoorloc: A new multi-building and multi-floor database for wlan fingerprint-based indoor localization problems," in *2014 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2014, pp. 261–270.
[23] H. L. E. S. Lohan, J. Torres-Sospedra *et al.*, "Wi-fi crowdsourced fingerprinting dataset for indoor positioning," *Data*, vol. 2, no. 4, p. 32, 2017.
[24] X. L. H. Zhu, L. Cheng *et al.*, "Neural-network-based localization method for wi-fi fingerprint indoor localization," *Sensors*, vol. 23, no. 15, p. 6992, 2023.
[25] R. P. N. Singh, S. Choe, "Machine learning based indoor localization using wi-fi rssi fingerprints: An overview," *IEEE Access*, vol. 9, pp. 127 150–127 174, 2021.
[26] B. Y. M. T. Hoang, Y. Zhu *et al.*, "A soft range limited k-nearest neighbors algorithm for indoor localization enhancement," *IEEE Sensors Journal*, vol. 18, no. 24, pp. 10 208–10 216, 2018.
[27] N. L. S. Zhang, J. Guo *et al.*, "Improving wi-fi fingerprint positioning with a pose recognition-assisted svm algorithm," *Remote Sensing*, vol. 11, no. 6, p. 652, 2019.
[28] Z. Huang, M. Valkama, J. Zhang, M. Xu, C. Yin and M. Guan, "WiLoc: Encoding-based WiFi Indoor Localization," in *Proc. 14th Int. Conf. on Indoor Positioning and Indoor Navigation (IPIN)*, Kowloon, Hong Kong, 2024, pp. 1–6.
[29] X. D. M. T. Hoang, B. Yuen *et al.*, "Recurrent neural networks for accurate rssi indoor localization," *IEEE Internet of Things Journal*, vol. 6, no. 6, pp. 10 639–10 651, 2019.
[30] H. R. M. Abbas, M. Elhamshary *et al.*, "Wideep: Wifi-based accurate and robust indoor localization system using deep learning," in *2019 IEEE International Conference on Pervasive Computing and Communications*. IEEE, 2019, pp. 1–10.
[31] C. X. X. Song, X. Fan *et al.*, "A novel convolutional neural network based indoor localization framework with wifi fingerprinting," *IEEE Access*, vol. 7, pp. 110 698–110 709, 2019.
[32] X. W. F. Qin, T. Zuo, "Ccpos: Wifi fingerprint indoor positioning system based on cdae-cnn," *Sensors*, vol. 21, no. 4, p. 1114, 2021.
[33] J. K. A. Alitaleshi, H. Jazayeriy, "Ea-cnn: A smart indoor 3d positioning scheme based on wi-fi fingerprinting and deep learning," *Engineering Applications of Artificial Intelligence*, vol. 117, p. 105509, 2023.
[34] M. Zhang, Z. Fan, R. Shibasaki and X. Song, "Domain Adversarial Graph Convolutional Network Based on RSSI and Crowdsensing for Indoor Localization," *IEEE Internet of Things Journal*, vol. 10, no. 15, pp. 13662–13672, 2023.
[35] L. Ayinla, A. A. Aziz, M. Drieberg, M. Susanto, A. Tumian and M. Yahya, "An Enhanced Deep Neural Network Approach for WiFi Fingerprinting-Based Multi-Floor Indoor Localization," *IEEE Open Journal of the Communications Society*, vol. 6, pp. 560–575, 2025.
[36] N. Yu, X. Wang and Y. Tian, "WiNCDN: A Novel WiFi-Assisted Non-Cooperative Indoor Localization System via Dropout-Based Neural Network," *IEEE Communications Letters*, vol. 29, no. 2, pp. 353–357, Feb. 2025.
[37] Y. H. K. B. Lee, K. M. Park *et al.*, "Hybrid approach for indoor localization using received signal strength of dual-band wi-fi," *Sensors*, vol. 21, no. 16, p. 5583, 2021.
[38] P. C. D. J. Suroso, F. Y. M. Adiyatma *et al.*, "Fingerprint database enhancement by applying interpolation and regression techniques for iot-based indoor localization," *Emerging Science Journal*, vol. 4, pp. 167–189, 2022.
[39] S. H. H. Park, C. Laoudias *et al.*, "Dropout autoencoder fingerprint augmentation for enhanced wi-fi ftm-rss indoor localization," *IEEE Communications Letters*, vol. 27, no. 7, pp. 1759–1763, 2023.
[40] S. M. Yu, K. Han, J. Park, S.-L. Kim and S.-W. Ko, "Combinatorial Data Augmentation: A Key Enabler to Bridge Geometry- and Data-Driven WiFi Positioning," *IEEE Transactions on Mobile Computing*, vol. 24, no. 1, pp. 306–320, Jan. 2025.
[41] A. C. W. Njima, M. Chafii *et al.*, "Indoor localization using data augmentation via selective generative adversarial networks," *IEEE Access*, vol. 9, pp. 98 337–98 347, 2021.
[42] F. G. W. Qian, F. Lauri, "Supervised and semi-supervised deep probabilistic models for indoor positioning problems," *Neurocomputing*, vol. 435, pp. 228–238, 2021.
[43] N. Yoon, W. Jung and H. Kim, "DeepRSSI: Generative Model for Fingerprint-Based Localization," *IEEE Access*, vol. 12, pp. 66196–66213, May 2024.
[44] S. T. J. Torres-Sospedra, R. Montoliu *et al.*, "Comprehensive analysis of distance and similarity measures for wi-fi fingerprinting indoor positioning systems," *Expert Systems with Applications*, vol. 42, no. 23, pp. 9263–9278, 2015.
[45] M. W. D. P. Kingma, "Auto-encoding variational bayes," arXiv preprint arXiv:1312.6114, 2013.
[46] X. Y. K. Sohn, H. Lee, "Learning structured output representation using deep conditional generative models," in *Advances in Neural Information Processing Systems*, vol. 28, 2015.
[47] J. N. D. Quezada-Gaibor, J. Torres-Sospedra *et al.*, "Surimi: Supervised radio map augmentation with deep learning and a generative adversarial network for fingerprint-based indoor positioning," in *2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*. IEEE, 2022, pp. 1–8.
[48] M. I. Mahali, J.-S. Leu, N. A. S. Putro, S. W. Prakosa and C. Avian, "DeepFuzzLoc Positioning: A Unified Fusion of Fuzzy Clustering and Deep Learning for Scalable Indoor Localization Using Wi-Fi RSSI," in *Proc. Joint 13th Int. Conf. on Soft Computing and Intelligent Systems and 25th Int. Symp. on Advanced Intelligent Systems (SCIS & ISIS)*, IEEE, 2024, pp. 1–6.
[49] E. L. J. Cha, "A hierarchical auxiliary deep neural network architecture for large-scale indoor localization based on wi-fi fingerprinting," *Applied Soft Computing*, vol. 120, p. 108624, 2022.
[50] J.-W. Jang and S.-N. Hong, "Indoor localization with wifi fingerprinting using convolutional neural network," in *2018 Tenth International Conference on Ubiquitous and Future Networks (ICUFN)*. IEEE, 2018, pp. 753–758.

Shixun Wu was born in Hubei, China. He received the B.S. and M.S. degrees in applied mathematics in 2006 and 2009, respectively. He received the Ph.D. degree in 2012 from the Department of Electrical and Computer Engineering, Central China Normal University, China. He is currently an Associate Professor at the College of Information Science and Engineering, Chongqing Jiaotong University, and also serves as the Vice Dean of the Department of Communication Engineering. His research interests include wireless localization, neural network localization, and wireless communication.

Xinrui Zeng was born in 2000. He is currently pursuing the M.S. degree in Computer Science and Technology at Chongqing Jiaotong University. His research interests include Wi-Fi indoor positioning and deep learning applications.

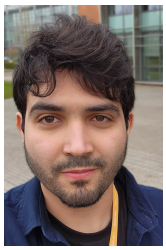
Miao Zhang received the B.Sc. degree in Optical Information Science and Technology from Guizhou University, China, the M.Sc. degree in Communications and Signal Processing from the University of Newcastle upon Tyne, UK, and the Ph.D. degree from the University of York, UK, in 2011, 2015, and 2020, respectively. He is currently an Associate Professor with the School of Information Science and Engineering, Chongqing Jiaotong University. His research interests include convex optimization, intelligent reflecting surface-aided wireless networks, physical layer security, and machine learning in wireless communications.



Kanapathippillai Cumanan (M'10, SM'19) Dr. Cumanan received the BSc degree with first class honors in Electrical and Electronic Engineering from the University of Peradeniya, Sri Lanka in 2006 and subsequently obtained his PhD degree in Signal Processing for Wireless Communications from Loughborough University, Loughborough, UK, in 2009.

He is currently a Professor of Wireless Communications at the School of Physics, Engineering and Technology, University of York, UK. Prior to this he served as an Assistant Lecturer at the Department of Electrical and Electronic Engineering, University of Peradeniya, Sri Lanka from January 2006 to August 2006. Between January 2010 and March 2012, he held a Research Associate position at Loughborough University, UK, followed by a similar role at Newcastle University, Newcastle upon Tyne, UK from March 2012 to November 2014. He joined as lecturer at University of York, UK in November 2014. In 2011, he was also an Academic Visitor at the Department of Electrical and Computer Engineering, National University of Singapore.

His research interests include non-orthogonal multiple access (NOMA), cell-free massive MIMO, physical layer security, cognitive radio networks, convex optimization techniques and resource allocation techniques. He has published more than 100 journal articles and conference papers which have collectively received more than 4500 Google Scholar citations. Dr. Cumanan was the recipient of an Overseas Research Student Award Scheme (ORSAS) from Cardiff University, Wales, UK, where he was a research student between September 2006 and July 2007.



Abdulhamed Waraiet (Member, IEEE) Received the BSc degree in Electrical and Electronics Engineering from Near East University, Northern Cyprus, in 2017, and the MSc degree in Signal Processing and Communications from the University of Edinburgh, UK, in 2019. He obtained his PhD in Electronic Engineering from the University of York in 2024 and has been working as a Postdoctoral Research Associate at the same institution since then. He was recently awarded the K.M. Stott Prize for Research Excellence. His current research interests include machine learning-based resource allocation algorithms for wireless communication systems.

Zheng Chu (Member, IEEE) is an Assistant Professor with the Department of Electrical and Electronic Engineering, University of Nottingham Ningbo China. Prior to this, he held positions at the University of Surrey from 2017 to 2024, and at Middlesex University from 2016 to 2017. He received the M.Sc. and Ph.D. degrees from Newcastle University, Newcastle-upon-Tyne, UK, in 2012 and 2016, respectively.

His current research interests include 6G, IoT networks, Artificial Intelligence (AI)-driven future networks, space-air-ground-sea integrated networks (SAGSIN), smart mobility, and transportation systems. He received the Exemplary Reviewer award from *IEEE Transactions on Communications* in 2022, and Best Paper Awards from IEEE/CIC UCOM, IEEE ICCT, and EAI CHINACOM in 2024.

Kai Xu was born in 1970. He received the M.S. degree in Control Theory and Control Engineering from Chongqing University in 2006. From 1993 to 2003, he served as a Senior Engineer at the Xichang Satellite Launch Center. Since 2008, he has been a Professor at the College of Information Science and Engineering, Chongqing Jiaotong University. He is also an expert commissioner of the Chongqing Science and Technology Commission and Chongqing Construction Commission. His research interests include fuzzy control, adaptive control, and intelligent algorithms.