mathematics

MDPI

Article

P-CA: Privacy-Preserving Convolutional Autoencoder-Based Edge–Cloud Collaborative Computing for Human Behavior Recognition

Haoda Wang, Chen Qiu, Chen Zhang, Jiantao Xu and Chunhua Su *

Graduate School of Computer Science and Engineering, The University of Aizu, Aizuwakamatsu 965-8580, Fukushima Prefecture, Japan; d8242105@u-aizu.ac.jp (H.W.); d8222103@u-aizu.ac.jp (C.Q.); d8252109@u-aizu.ac.jp (C.Z.); d8252108@u-aizu.ac.jp (J.X.)
* Correspondence: chsu@u-aizu.ac.jp

Abstract: With the development of edge computing and deep learning, intelligent human behavior recognition has spawned extensive applications in smart worlds. However, current edge computing technology faces performance bottlenecks due to limited computing resources at the edge, which prevent deploying advanced deep neural networks. In addition, there is a risk of privacy leakage during interactions between the edge and the server. To tackle these problems, we propose an effective, privacy-preserving edge–cloud collaborative interaction scheme based on WiFi, named P-CA, for human behavior sensing. In our scheme, a convolutional autoencoder neural network is split into two parts. The shallow layers are deployed on the edge side for inference and privacy-preserving processing, while the deep layers are deployed on the server side to leverage its computing resources. Experimental results based on datasets collected from real testbeds demonstrate the effectiveness and considerable performance of the P-CA. The recognition accuracy can maintain 88%, although it could achieve about 94.8% without the mixing operation. In addition, the proposed P-CA achieves better recognition accuracy than two state-of-the-art methods, i.e., FedLoc and PPDFL, by 2.7% and 2.1%, respectively, while maintaining privacy.

Keywords: human behavior recognition; edge computing; privacy preservation; deep learning

MSC: 68T07

Citation: Wang, H.; Qiu, C.; Zhang, C.; Xu, J.; Su, C. P-CA: Privacy-Preserving Convolutional Autoencoder-Based Edge–Cloud Collaborative Computing for Human Behavior Recognition. *Mathematics* **2024**, *12*, 2587. https://doi.org/10.3390/math12162587

Academic Editor: Ke-Lin Du

Received: 29 June 2024
Revised: 20 August 2024
Accepted: 21 August 2024
Published: 21 August 2024

1. Introduction

Human behavior recognition (HBR), empowered by artificial intelligence (AI), has spawned extensive applications in smart worlds [1,2]. For HBR, the accurate acquisition of target behavior information through electronics has attracted significant attention in various scenarios [3,4] such as AR-based real-scene navigation. Existing technologies, such as satellite positioning systems (e.g., GPS) and radio-frequency identification, generally require targets to carry specific equipment or electronic tags. This device-based localization technology is not applicable in certain scenarios [5,6], such as intruder detection under intelligent security or elderly health monitoring under a smart home, etc. Some device-free sensing (DFS) technologies, which rely on computer vision and infrared, face many limitations in practical applications due to the need for light and obstacle-free conditions.

Among these technologies, WiFi-based DFS offers advantages such as a low cost, fast transmission rates, and fewer condition limitations, such as smoke, walls, and light [7,8]. These merits make it promising for ambient intelligence in areas such as real-time tracking and monitoring, and health care for the elderly or patients [9,10], etc. In a WiFi-based HBR system, channel state information (CSI) commonly contains behavior information of a target [11,12]. To achieve high recognition accuracy, machine learning algorithms are widely used in CSI-based DFS systems. Among them, deep learning algorithms are

especially appealing because of their strong capabilities in data processing and feature extraction, which are crucial for accurate sensing [13,14].

However, a significant challenge is that most existing deep learning-based DFS technologies have limited generalization ability in changing environments. To tackle this issue, several methods have been proposed from different perspectives. Zhang et al. [15] proposed a phase decomposition method to extract multi-path phases from CSI as fingerprints for the location recognition of a target. Li et al. [16] designed a deep neural network-based domain adaptation algorithm by conducting fine-grained alignment, which enforces the target domain to align with its corresponding part of the source domain. This method improved the performance of inference on the target domain data. Most of the existing work assumes that the edge side has sufficient computational resources to support high-performance deep learning models. However, in practical HBR systems, the computational resources of the edge side are often too limited to deploy complex models.

Benefiting from recent developments in intelligent electronics and integrated terminals, cloud computing and edge computing have significantly advanced the field of HBR. However, conventional cloud computing architecture suffers from serious problems. As shown in Figure 1, limited by insufficient computing power at the edge, raw data need to be uploaded to the cloud for processing and analysis. This cloud AI mode is plagued by several severe issues, such as privacy leakage risks, constrained intelligence at the edge for HBR, and high communication overhead [10,17]. This brings challenges to the development of HBR applications. A primary challenge for current deep learning-based HBR is preserving the behavior information of the targets.

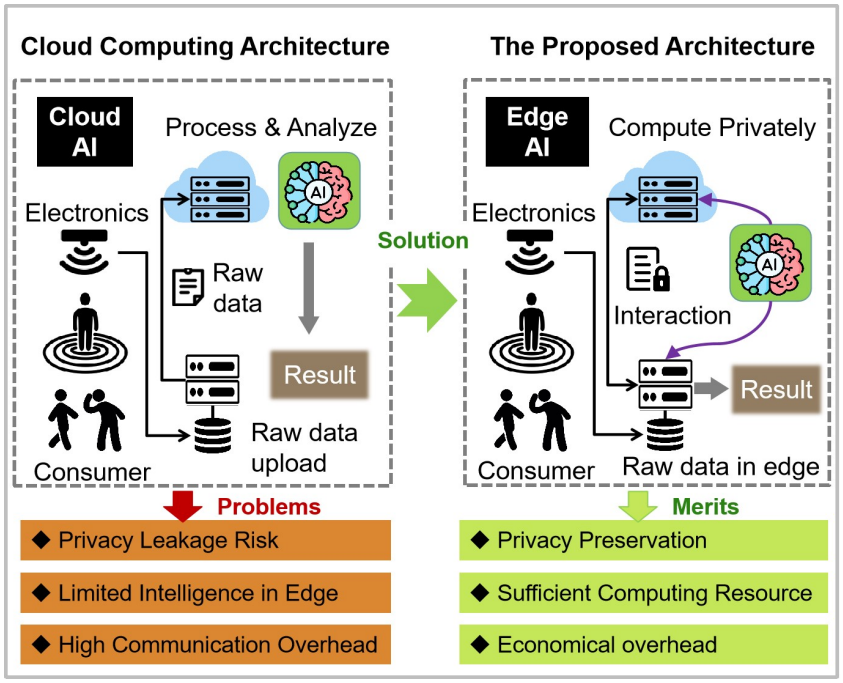


Figure 1. Challenges and the proposed solutions of shifting from cloud computing to edge computing architecture.

Since the monitoring system continuously records the user's activities, there is a risk of personal information leakage, which leads to privacy issues. Ensuring the privacy protection of user information has become a critical topic [18,19]. Existing research on privacy protection related to HBR technology, particularly in location recognition, has primarily focused on device-based methods like GPS [20,21]. Users' location information, such as device ID, coordinates, or motion trajectory, can be decoded in real time and uploaded in small sizes by their carried devices. Thus, user privacy can be safeguarded by implementing data encryption or mixing puppet information on the device side. However, the system

architecture of DFS technology based on wireless sensor networks is different. Users do not need to carry any electronic device, and their behavioral information is recorded in public wireless signals, leading to inapplicability of the aforementioned privacy protection schemes. Adding noise or blurring the signals to secure them (even wireless signals) is not foolproof; the original signals could be reconstructed using generative adversarial network-based attack models [22,23]. Additionally, there is a risk that performance might be degraded by the randomly added noise. Some recent studies report the effective federated learning methods for privacy-preserving DFS [24,25]. However, large number of signals need to be efficiently processed and uploaded to complex machine learning models for perceptual training. The large-scale data lead to high processing costs and computational burden on the device side. The limited computational power at the edge makes it challenging to support these federated learning-based schemes. Therefore, it is necessary to explore specific privacy protection mechanisms for the DFS system.

To address the aforementioned problems of HBR, we propose a privacy-preserving edge–cloud framework, named Privacy-Preserving Convolutional Autoencoder (P-CA). The P-CA prevents real behavior information leakage during data transmission while utilizing resource-abundant servers. On the one hand, unlike the previously mentioned methods, we first propose a data-mix scheme to generate datasets by mixing the amplitude features of WiFi signals. We then design a convolutional autoencoder neural network (CANN) for unsupervised training and feature extraction. After the training procedure, the CANN model can be saved and used to infer the target's behavior. The backbone of the CANN is strategically deployed to utilize high computing resources in a cloud server, while lightweight components of the CANN model are deployed on the edge side for efficient inference.

On the other hand, based on the theory of Mixup [26], we introduce mixture and de-mixture operations to prevent information leakage during client–server transmission. This approach ensures that an attacker can only steal the mixed information, not the original data or location. Figure 1 demonstrates the advantages of our proposed P-CA for HBR, comparing it with the cloud computing architecture and highlighting our innovative edge–cloud architecture.

The major contributions of our study are summarized as follows:

- A privacy-preserving edge–cloud interaction architecture, i.e., P-CA, for indoor WiFi-based human behavior recognition is proposed. Under this architecture, intelligent reasoning capabilities at the edge can be greatly improved by privately utilizing sufficient computing resources of the server.
- A three-layer CANN architecture together with an effective algorithm to learn its parameters is designed. In the testing mode, the trained model can automatically output the behavior inference of DFS targets based on CSI features.
- The privacy-preserving and behavior recognition ability of the proposed P-CA is verified on real-world datasets.

The remainder of this paper is organized as follows. Section 2 introduces the pioneering related works. Section 3 demonstrates the proposed algorithm. Section 4 presents the experiments and performance evaluation. Finally, Section 5 concludes this work.

2. Related Work

This section provides a summary of the literature on the development of HBR, specifically focusing on consumer localization and activity recognition. Subsequently, we briefly discuss the research related to edge computing for the aforementioned location and activity recognition.

2.1. Device-Free Human Behavior Recognition

Device-free HBR can be viewed as a target localization and activity recognition task within the designated monitoring area. Compared with other technologies, such as camera-based HBR, WiFi-based HBR offers advantages including a low cost, enhanced privacy,

and fewer condition limitations. However, due to signal interference, such as multi-path variations and environmental noise, achieving effective HBR remains a challenging task. To ensure accurate target localization and activity recognition, numerous studies have focused on exploring high-performance deep learning methods.

Liu et al. [27] conducted six activities of recognition, such as picking and returning an item on a shelf, based on hand movements and body orientation to analyze consumer preferences. Tom et al. [28] studied the daily activity patterns of consumers and their energy demand for reducing energy consumption. Li et al. [29] modeled fingerprint localization as a subspace matching problem and proposed a Siamese CNN to extract features from signals and infer the position of the target by comparing the similarity. To improve the localization accuracy and generality, Zhang et al. [30] proposed an attention-augmented residual neural network, which exhaustively utilized both local information and global context in CSI. For robust activity recognition, Wang et al. [31] employed generative adversarial networks (GANs) for data generation and designed CNNs for human activity classification. Yang et al. [32] designed a cross-model supervision method based on deep learning by mapping wireless signal information to accurate computer vision human pose landmarks. For effective spatial information exploration, Zhou et al. [33] designed a shared convolution mechanism and a Transformer module to map the CSI of WiFi signals to landmarks of human poses for movement estimation.

In summary, the aforementioned work has laid a foundation for further HBR research. Most of the existing work assumes that the edge side has sufficient computational resources to support high-performance deep learning models. Whereas, in practical HBR systems, the computational resources at the edge side are often too limited to deploy complex models effectively. Although many methods have been proposed to achieve high accuracy using complex deep neural networks, research on HBR in edge devices remains insufficiently explored.

2.2. Edge Computing for Human Behavior Recognition

Edge computing has increasingly become prominent in recent years within the field of target behavior recognition, including location and activity recognition [34–36]. Kwon et al. [36] deployed a tensor processing unit (TPU)-enabled edge computing camera system in the ceiling of a room and designed a multi-person detection algorithm to run on the edge TPU for real-time pose estimation of the targets. This system ensures the privacy of individuals and reduces data transmission/storage. Wu et al. [37] proposed a personalized federated learning framework for human activity recognition within a cloud–edge architecture in the Internet of Things (IoT) environment, leveraging edge computing for rapid processing and minimal latency. Narayana et al. [38] used a thermopile in conjunction with a PIR sensor for location estimation. Since a low-resolution thermopile array only provides the gait of the target instead of the actual image, they believe that the combination of the PIR sensor and thermopile is privacy-preserving. Cominelli et al. [39] designed a randomization approach of CSI. It prevents unauthorized localization by adding randomly generated measurements to the CSI signal, such as random peaks and random phase jumps. Shi et al. [40] designed self-powered triboelectric mats and, based on them, a smart floor monitoring system was proposed for location and activity sensing. This system was designed by running a deep learning-based model on instant sensory data.

Although the previous research on edge computing is considered to preserve privacy at the network edge, some data still need to be transmitted between the edge and the cloud. Moreover, deploying deep learning models with high resource requirements on edge devices remains challenging.

3. The Proposed Algorithm

3.1. Preliminary

Figure 2 shows an illustration of the indoor HBR system model based on CSI. The detection area is monitored by a transmitter and receiver pair, both of which are MiniPCs

equipped with an Intel 5300 WiFi card. The two Intel 5300 WiFi cards are equipped with omnidirectional antennas. Specifically, the CSI signals are collected from the Intel 5300 card using the open-source CSITool [41]. Note that the received signals between the transmit antennas and the receive antennas are affected by the target and the environment, such as reflections from the target, the ceiling, or the furniture. As shown in this figure, the monitoring area is divided into a series of grids, and each point within one grid is considered a potential location, i.e., reference point (RP). Considering that the target in different locations or activities may lead to various patterns in the collected CSI signals, the ground truth is obtained by manually labeling the signals collected with the target in each grid.

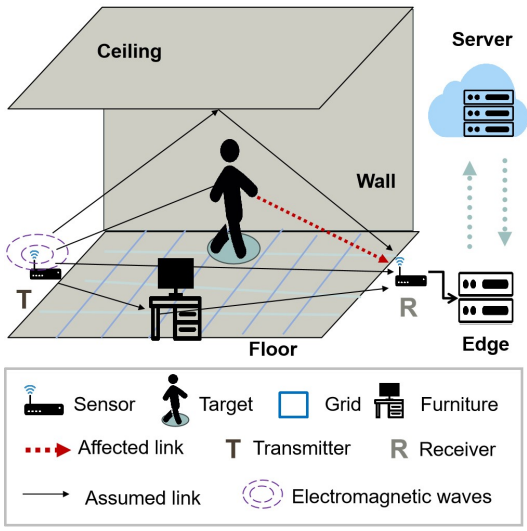


Figure 2. Illustration of CSI-based indoor HBR system model. It showcases how electromagnetic waves emitted by a transmitter (T) interact with the environment, causing variations in the signal. These signals, affected by human behavior, are captured by a receiver (R) for analysis. The system is designed to detect and interpret human behavior based on the changes in the CSI, which are processed and analyzed at the edge and server levels.

3.2. Mixture Strategy

The mixture strategies designed in this work are inspired by Mixup [26]. Mixup constructs new samples by mixing a pair of samples using coefficients, as well as their corresponding labels. Different from Mixup performed for data augmentation, in this paper, the mixing operation is mainly for protecting information during data transmission. The mixture operation is shown in Figure 3, and denoted as

$$\hat{\boldsymbol{x}} = \lambda_1 \boldsymbol{x}_i + (1 - \lambda_1)\boldsymbol{x}_j, \tag{1}$$

$$\hat{\boldsymbol{y}} = \lambda_1 \boldsymbol{y}_i + (1 - \lambda_1)\boldsymbol{y}_j, \tag{2}$$

where $i \neq j$, λ_1 is the coefficient for mixing two samples randomly sampled from $B(\alpha, \beta) \in (0, \infty)$. $\boldsymbol{x}$ and $\boldsymbol{y}$ denote the raw input sample and the corresponding label. $\hat{\boldsymbol{x}}$ and $\hat{\boldsymbol{y}}$ denote the mixed ones. As shown in this equation, the signals are linearly mixed by element-wise addition.

In this work, we constructed the entire dataset by employing three different targets. As is common, each dataset of one target is denoted as one domain. In this paper, random mixing is adopted for the mixture operation. This means that the two samples for the mixture are randomly selected from the datasets of different targets, regardless of their classes or domains. As shown in Equations (1) and (2), the random mixed samples are obtained by four kinds of mixtures, including (1) $\boldsymbol{y}_i^C \neq \boldsymbol{y}_j^C$ and $\boldsymbol{y}_i^D = \boldsymbol{y}_j^D$, (2) $\boldsymbol{y}_i^C = \boldsymbol{y}_j^C$ and $\boldsymbol{y}_i^D \neq \boldsymbol{y}_j^D$, (3) $\boldsymbol{y}_i^C \neq \boldsymbol{y}_j^C$ and $\boldsymbol{y}_i^D \neq \boldsymbol{y}_j^D$, and (4) $\boldsymbol{y}_i^C = \boldsymbol{y}_j^C$ and $\boldsymbol{y}_i^D = \boldsymbol{y}_j^D$, where C denotes classes and D denotes domains of the sample.

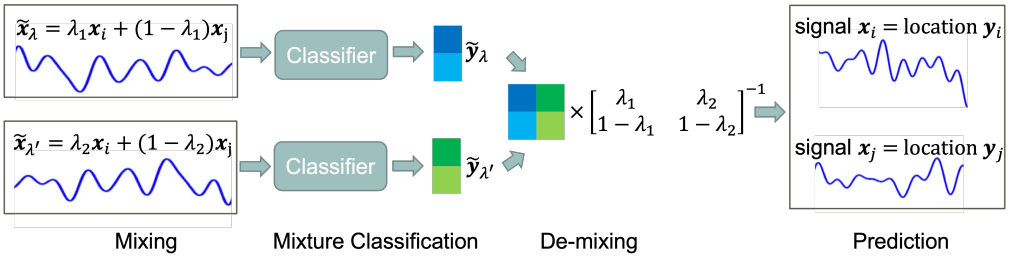


Figure 3. Illustration of the data or feature mixture strategy used for signal prediction. It shows the end-to-end process of mixing, classifying, and de-mixing signals to achieve accurate predictions.

Note that in addition to mixing the samples we extend the mixture operation to intermediate feature maps extracted from shallow layers of the CANN. This means that the outputs from the first several layer(s) would be utilized in the mixture process. Given that these intermediate feature maps are projections of the input data, they may already have extracted important information. Thus, this approach not only improves mixtures for subsequent feature extraction layers but also protects the information in the raw data.

3.3. Framework of the Proposed Privacy-Preserving Convolutional Autoencoder (P-CA)

Figure 4 illustrates the framework of the proposed P-CA approach. It consists of three parts: data collection and data preprocessing, privacy-preserving model training, and human behavior inference.

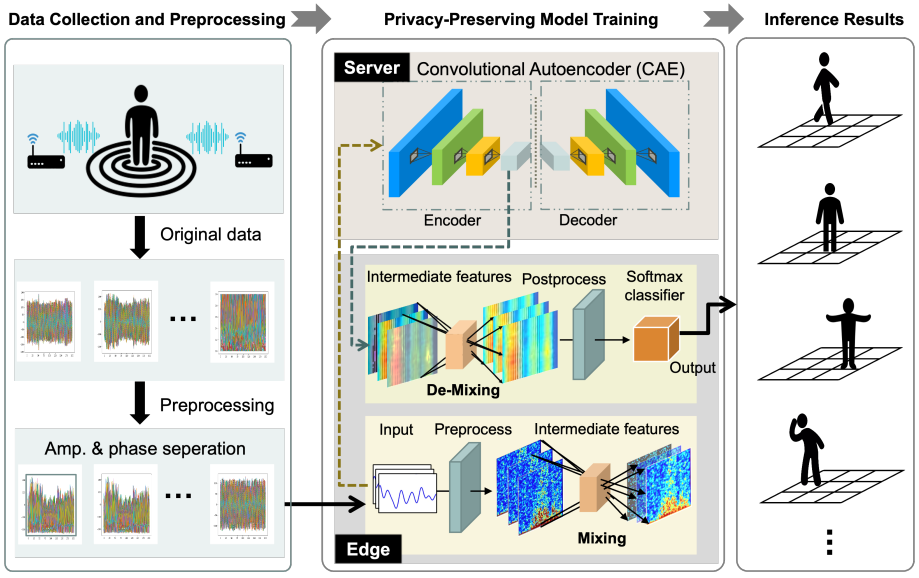


Figure 4. Framework of the proposed privacy-preserving convolutional autoencoder (P-CA) for human behavior recognition. It consists of three main stages: data collection and preprocessing, privacy-preserving model training, and results inference.

3.3.1. Data Collection

As demonstrated in Section 3.1, two MiniPCs served as the transceiver and receiver for CSI data collection. In an orthogonal frequency division multiplexing (OFDM) system, CSI is an attribute of the communication link. In detail, it reflects accurate multi-path information on subcarriers between the transmit antennas and receive antennas in an OFDM system. Generally, the CSI of a link, i.e., the CSI measurement, collected from the i_{th} transmit antenna and the j_{th} receive antenna carried by the k_{th} subcarrier can be described

as $H_{i,j,k}$. It contains the k_{th} subcarrier information for this link. The definition of the CSI measurement is as follows:

$$\begin{aligned} H_{i,j,k} &= |H_{i,j,k}|exp(-j\phi_{i,j,k}), \\ i \in \{1,2,\ldots,N_t\}, j \in \{1,2,\ldots,N_r\}, k \in \{1,2,\ldots,N_s\}. \end{aligned} \tag{3}$$

Here, N_t and N_r represent the number of antennas for the transmitter and receiver, respectively. N_s denotes the number of subcarriers in each radio beam according to the IEEE 802.11 a/g/n standard. $|H_{i,j,k}|$ and $exp(-j\phi_{i,j,k})$ denote the amplitude and phase information of the k_{th} subcarrier channel, respectively. Note that only the amplitude information is taken in this work.

During dataset acquisition, referencing the common CSI sequence of the WiFi signal collection mechanism [1,42], M CSI packets are collected at a fixed rate by having a target in different RPs. The collected packets of the u_{th} RP are $\boldsymbol{H}_u = (\boldsymbol{H}_u^1, \ldots, \boldsymbol{H}_u^m, \ldots, \boldsymbol{H}_u^M)$, where $m \in \{1,2,\ldots,M\}$ denotes the CSI packets and M represents the maximum value of data packets for an RP. $u \in \{1,2,\ldots,U\}$ denotes the index of an RP. In detail, the CSI measurement is collected from the data packet m of the u_{th} RP, named $\boldsymbol{H}_u^m$. Note that the value of each element in $\boldsymbol{H}_u^m$ represents the CSI measurement $H_{i,j,k}$, as defined in Equation (3).

$$\begin{aligned} \boldsymbol{H}_u^m = \Bigg(& \begin{bmatrix} H_{1,1,1}^{u,m} & \cdots & H_{1,1,N_s}^{u,m} \\ \vdots & \ddots & \vdots \\ H_{1,N_r,1}^{u,m} & \cdots & H_{1,N_r,N_s}^{u,m} \end{bmatrix}, \cdots, \begin{bmatrix} H_{i,1,1}^{u,m} & \cdots & H_{i,1,N_s}^{u,m} \\ \vdots & \ddots & \vdots \\ H_{i,N_r,1}^{u,m} & \cdots & H_{i,N_r,N_s}^{u,m} \end{bmatrix}, \\ & \cdots, \begin{bmatrix} H_{N_t,1,1}^{u,m} & \cdots & H_{N_t,1,N_s}^{u,m} \\ \vdots & \ddots & \vdots \\ H_{N_t,N_r,1}^{u,m} & \cdots & H_{N_t,N_r,N_s}^{u,m} \end{bmatrix} \Bigg) \end{aligned} \tag{4}$$

In this paper, $N_t = 1, N_r = 3, N_s = 30, M = 500$, and $U = 31$. Therefore, the size of each CSI data sample is 3×30.

3.3.2. Data Preprocessing

To ensure the accuracy and reliability of the CSI-based HBR model, preprocessing the collected CSI samples to minimize environmental noise is crucial. Note that each group of CSI samples contains environmental information, including environmental noise. Therefore, preprocessing these samples is essential before using them for multi-dimensional information recognition, to reduce environmental impacts. First, we collected CSI samples without any target in the detection area for approximately 30 s. Then, we calculated the average CSI measurements collected from these CSI packets, named $\bar{\boldsymbol{H}}_{vacant}$. Finally, the $\bar{\boldsymbol{H}}_{vacant}$ was subtracted from each $\boldsymbol{H}_u^m$ to reduce the influence of the environment: $\delta\boldsymbol{H}_u^m = \boldsymbol{H}_u^m - \bar{\boldsymbol{H}}_{vacant}$. In this paper, $\delta\boldsymbol{H}_u^m$ is used for training and testing.

For data preprocessing, we performed amplitude and phase separation, employing only amplitude information in this work. Additionally, background subtraction was also utilized to mitigate environmental influences.

3.3.3. Privacy-Preserving Model Training and Human Behavior Inference

The P-CA framework consists of the cloud server side and the edge side. In this work, privacy preservation is considered during edge–cloud data transmission and in scenarios involving a potentially malicious cloud server. The privacy-preserving model training involves secure information transmission between the edge and cloud, along with the model's unsupervised pre-training. Information protection is achieved through a mixing operation, which is illustrated in Section 3.2. The parameters of the encoder network can be obtained after pre-training of the CANN, and then, the target's behavior can be predicted by adding a softmax classifier. The local resource requirements can be reduced since only a few layers are trained on the edge side.

From Figure 4, the mixture and corresponding de-mixture operations are applied directly to the input data and classification outputs only on the edge side. Hence, potential attackers or the cloud server can only access the mixed information. For input data or the intermediate feature maps, together with their class labels, they are linearly mixed by element-wise addition with a pair of coefficients, as illustrated in Equations (1) and (2). Mixed data can hide original information because entropy increases when element values from the two samples are combined, therefore preserving privacy. Aware of the value of the pair of coefficients $\Lambda = [\boldsymbol{\lambda}; \boldsymbol{\lambda'}](\boldsymbol{\lambda} = [\lambda_1, 1-\lambda_1]^T, \boldsymbol{\lambda'} = [\lambda_2, 1-\lambda_2]^T)$, the true classification results can be obtained from the mixed labels, i.e., $\hat{\boldsymbol{y}}_\lambda, \hat{\boldsymbol{y}}_{\lambda'}$, through the de-mixing operation.

$$[\boldsymbol{y}_i, \boldsymbol{y}_j] = [\hat{\boldsymbol{y}}_\lambda, \hat{\boldsymbol{y}}_{\lambda'}]\Lambda^{-1}, \tag{5}$$

In addition, the summation coefficients are produced by the edge clients, and are only kept on the edge. Therefore, the transmitted data, feature maps, or correct classes cannot be recovered by the cloud or attackers.

3.3.4. Convolutional Autoencoder Neural Network (CANN)

The CANN leverages the merits of convolutional spatial feature extraction and unsupervised pre-training of the autoencoder. In the stage of unsupervised pre-training, features underlying the mixed data are extracted by reconstructing the input data.

After pre-training the CANN, the decoder network is removed. Then, several fully connected layers and a classifier are deployed following the encoder part for inference. Note that we utilize convolutional operations with a stride of 2 instead of pooling operations in this CANN architecture. In the unsupervised pre-training, mean square error is used to minimize the gap between the input signal and the recovered one. For the supervised training of the encoder network, the cross-entropy function is utilized. It computes the error between the estimated result $\boldsymbol{y}'$ and ground truth $\boldsymbol{y}$. The cost function is defined as

$$J(\boldsymbol{\theta}) = \frac{1}{S}\sum_{s=1}^{S} \boldsymbol{y}_s \cdot \log(\boldsymbol{y}'_s(\boldsymbol{\theta})), \tag{6}$$

where $\boldsymbol{\theta}$ denotes the trainable parameters in the encoder network, and $S = M \times U$ denotes the sum sample number. Other detail settings such as 'dropout', activation function, and optimization algorithm are taken from [43].

In addition, the majority of the computations are delegated to the cloud by deploying the main model on the server. With only a few layers of the CANN situated on the edge side, the computational load of the edge device is significantly reduced. Despite the mixing and de-mixing operations for privacy preservation, the main task of the few layers on the edge is to infer the behavior of the target. For the tightly resource-constrained edge device, fewer computations are beneficial for more effective and efficient inference.

4. Performance Evaluation

This section evaluates the performance of the P-CA approach using the real-world experimental dataset illustrated in Section 3.1. All the experiments are implemented in PyTorch 1.13.1 on a system with two GeForce GTX 1080 Ti GPUs and 64 GB memory.

4.1. Configuration of Experiments

The real-world experimental dataset for localization is collected in an apartment, and the monitoring area includes a living room and a bedroom. The transmitter and receiver are deployed at the opposite end of the living room. The transmit antennas and receive antennas are all placed 1.2 m above the ground. Specifically, we select 26 RPs in the living room and 5 RPs in the bedroom. The grid for each RP is 0.5 m $\times$ 0.5 m. The other experimental setup is for activity recognition, which is deployed in an indoor office room. The transmit antennas and receive antennas are placed 3 m apart in a line-of-sight condition. The activity involved a target moving and performing actions over a 20-s period.

Data description: In this paper, two groups of datasets are collected, including the location data group and the activity data group. To explore the influence of different targets, we employed three different targets to construct three sub-datasets for location recognition. The localization dataset is named L-dataset, and sub-datasets of each different target are named L-data1, L-data2, and L-data3. As mentioned, the L-dataset is collected in an apartment, which has 31 RPs in total. The samples are collected by the target standing in each RP for a fixed time period of approximately 30 s, and 500 samples at each RP are selected.

Different from the L-dataset, the data for activity recognition is from a public dataset [44], which is called the A-dataset in this paper. The A-dataset has a total of 417 samples, which contains seven classes, including bed, fall, pick up, run, sit down, stand up, and walk (https://github.com/ermongroup/Wifi_Activity_Recognition, accessed on 28 September 2023). Note that these samples were collected by six different volunteers. Based on our previous experiments, dividing the A-dataset into six sub-datasets by target leads to serious overfitting. This is because each sub-dataset contains too few samples. Therefore, we directly employ the whole A-dataset for training. For both the L-dataset and A-dataset, 80% of them are randomly selected for training and the remaining 20% are used for testing.

Compared methods: In this work, we compared the proposed P-CA with two state-of-the-art (SOTA) methods, i.e., Federated Localization (FedLoc) [24] and Privacy-Preserving Device-Free Localization (PPDFL) [25]. Both methods use a privacy-preserving edge–cloud interaction scheme based on federated learning, in addition to implementing HBR based on deep learning models. Both approaches use a privacy-preserving edge–cloud interaction scheme based on federated learning, in addition to implementing HBR based on deep learning models. To ensure the fairness of the comparison experiments, we conduct evaluation experiments on the same dataset.

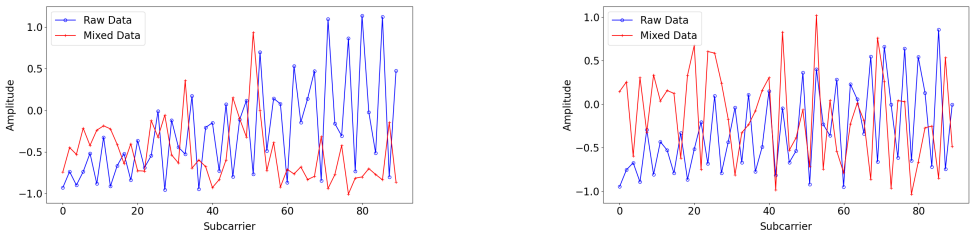
4.2. Performance of the Proposed Privacy-Preserving Convolutional Autoencoder (P-CA)

4.2.1. Effectiveness and Contribution of Mixture Strategies to the P-CA

In this part, we evaluate the effectiveness of two mixture strategies, i.e., mixing the raw data and mixing the intermediate feature maps.

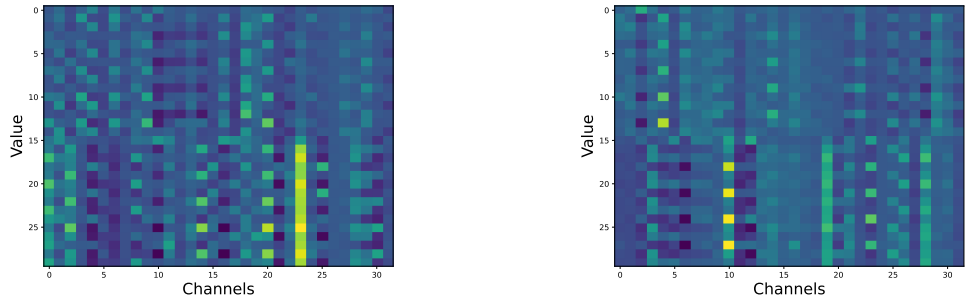
Figure 5 presents the visualization of a raw single sample and a mixed one. After the mixture operation, over 90% of the amplitude values in subcarriers are different from the raw data. Figure 6 shows the visualization of feature maps before and after mixture operations. Figure 6a denotes the raw feature map extracted from the shallow layer of the CANN. Here, this feature map means the output of the last layer of the CANN at the edge, and different colors in this figure denote different values. Figure 6b presents a visualization of the mixed feature maps. The results are obtained by randomly selecting two intermediate feature maps. After this mixture operation, privacy information in the raw feature maps could be hidden. The distribution and patterns have been changed after mixing, regardless of the data or feature maps. Figure 7 illustrates the clustering results of raw data and the mixing data. Through the operations of data and feature mixing, it becomes very hard to separate the data into correct classes (see Figure 7b,d). Since only the edge client holds the coefficient values for mixing and de-mixing, the cloud server has no access to the raw data or features. The third party thus cannot infer the correct behavior, i.e., location and activity, of the target.

To explore the influence of mixing strategies, we implemented a series of experiments on the proposed P-CA with and without mixing operations. Table 1 summarizes the corresponding comparison results. In this table, P-CA (w/o mix) denotes training and testing the raw data without mixing strategies. P-CA (data mix) represents the random mixing of the training data before transferring it to the cloud. P-CA (feature mix) denotes transferring the mixed feature maps to the cloud for training and testing.



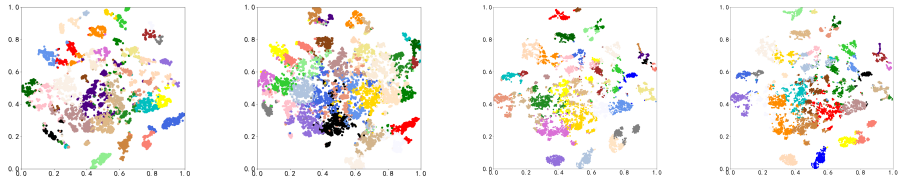
(**a**) Visualization of sample 1 (**b**) Visualization of sample 2

Figure 5. Comparison of the data samples before and after data mixing operation.



(**a**) Raw feature map (**b**) Mixed feature map

Figure 6. Comparison of raw feature map learned by the P-CA without mixture and the feature map after the mixture operation. (**a**) shows a visualization of a raw feature map extracted from the shallow layer of the CANN, and (**b**) presents a random mixed feature map.



(**a**) Raw data cluster (**b**) Mixed data cluster (**c**) Feature maps cluster (**d**) Mixed feature maps cluster

Figure 7. Comparison of raw data cluster and mixed data cluster. (**a**) Visualization of the cluster results on raw data. (**b**) Visualization of the cluster results on mixed data. (**c**) Visualization of the cluster results on raw feature maps. (**d**) Visualization of the cluster results on mixed feature maps.

Table 1. Localization accuracy comparison of the proposed P-CA without mixture, with data mixing, and with feature mixing.

Training Data	**Testing Data**	**P-CA (w/o mix)**	**P-CA (data mix)**	**P-CA (feature mix)**
L-data1	L-data1	94.8%	78.9%	88.0%
L-dataset	L-dataset	93.8%	77.4%	79.6%
L-data1	L-data2 + L-data3	16%	45.3%	73%

From the results of the first row in Table 1, 80% of the L-data1 are employed as training data and 20% as testing data. The P-CA without mixture achieved a location recognition accuracy of 94.8%. After mixing, whether mixing data or features, the localization accuracy decreased. However, P-CA with the feature mixing performed better. Compared to data mixing, it achieved an accuracy of 88.0%, which is only 5% less than P-CA without mixture.

4.2.2. Performance of the Edge–Cloud Interactive P-CA for HBR

In this part, the performance of P-CA is evaluated based on three datasets collected from different targets for HBR, including human localization and activity recognition.

Human localization: The distributions of samples collected from different targets are distinct, even within the same location. Therefore, to evaluate the performance of P-CA with various mixing strategies, we extended the experiment to the L-dataset. The L-dataset includes L-data1, L-data2, and L-data3. As the results in the second row of Table 1 show, the accuracies of P-CA (without mix), P-CA (data mix), and P-CA (feature mix) all decline. However, the difference between P-CA (without mix) and P-CA (data mix) remains at approximately 5%. Additionally, P-CA (feature mix) outperforms P-CA (data mix). These results suggest that features extracted by the shallow layers provide better mixtures for the main model, thereby enhancing inference accuracy.

To evaluate the generalization ability of P-CA, we trained the model using L-data1 and tested it on L-data2 and L-data3. The accuracy obtained by P-CA (without mix) is 16%, indicating a failure in inference. Although P-CA (data mix) achieves a better accuracy of 45.3%, it is still below 50%. P-CA (feature mix) achieves the highest localization accuracy of 73%. These results suggest that mixing feature maps not only prevents information leakage but also maintains higher recognition accuracy. Additionally, employing the feature mixing strategy improves the generalization ability of the model. For samples collected from new situations not encountered during training, the feature mixing operation helps the model achieve higher inference accuracy.

Since P-CA (feature mix) demonstrates the best performance, additional experiments were conducted to evaluate its effectiveness on different datasets. Figure 8 shows the training process of P-CA (feature mix) on L-data1 and L-data1/2. L-data1/2 denotes a combination of L-data1 and L-data2. To assess the stability of the training process, each experiment was repeated 30 times. The bold lines in the figure represent the average accuracy and loss, while the shaded areas indicate the corresponding variances. The small size of the shaded areas suggests that the 30 training iterations are stable.

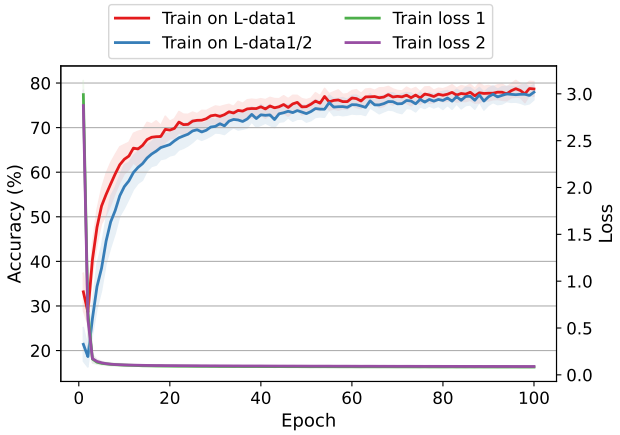


Figure 8. Training procedure of P-CA (feature mix) on L-data1 and L-data2/3.

Figure 9 illustrates the cumulative distribution function (CDF) of the 30 testing accuracies of P-CA (feature mix) on L-data2/3 and L-data3. Although both groups of testing accuracies range between 70% and 86%, the results on L-data3 are better. Specifically, the average testing accuracy on L-data3 is 78.6%, which is 5.6% higher than that on L-data2/3. This indicates that P-CA trained on L-data1/2 outperforms the model trained on L-data1 alone. Therefore, combining different datasets, i.e., different targets, may enhance the performance of P-CA.

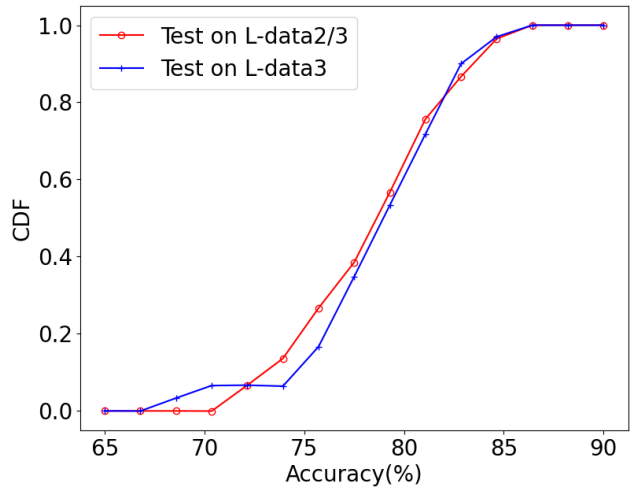


Figure 9. Testing accuracy of P-CA (feature mix) on L-data2/3 and L-data3. Note that testing on L-data2/3 indicates P-CA was trained on L-data1, and testing on L-data3 indicates P-CA was trained on L-data1/2.

Human activity recognition: To enhance the evaluation of human behavior, we conducted experiments on human activity recognition in addition to location recognition. Figure 10 presents the cumulative distribution function (CDF) plot of the accuracy for 30 activity recognition tests, comparing P-CA (data mix) and P-CA (feature mix). The accuracy of P-CA (data mix) ranges from 85.0% to 95.0%, while the accuracy of P-CA (feature mix) ranges from 80.0% to 95.0%. This suggests that P-CA (data mix) demonstrates more stable performance in activity recognition tasks. However, since the majority of accuracies for both mixing strategies are around 90%, it can be concluded that data mixing and feature mixing achieve comparable results.

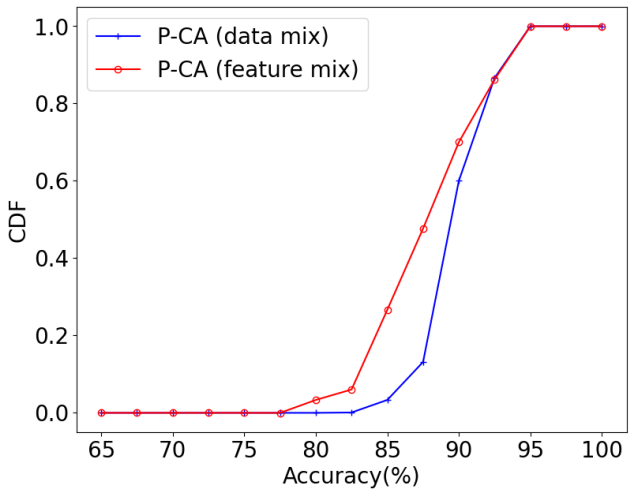


Figure 10. Testing accuracy of P-CA (data mix) and P-CA (feature mix) on activity recognition. Note that the CDF plot is based on tests conducted 30 times.

Table 2 summarizes the average training and testing accuracy for activity recognition using P-CA (w/o mix), P-CA (data mix), and P-CA (feature mix). Similar to its performance in location recognition, the implementation of mixing strategies in P-CA achieved privacy preservation with a slight reduction in accuracy. Specifically, P-CA (feature mix) attained an accuracy of 87.8%, which is only 1.6% lower than that of P-CA (data mix). Additionally, P-CA (data mix) achieved an accuracy of 89.4%, which is comparable to the accuracy of P-CA (w/o mix). These results suggest that the proposed P-CA with mixing strategies effectively balances privacy preservation and accuracy.

Table 2. Activity recognition accuracy comparison of the proposed P-CA without mixing, with data mixing, and with feature mixing.

	P-CA (w/o mix)	**P-CA (data mix)**	**P-CA (feature mix)**
Training accuracy	100%	89.7%	87.9%
Testing accuracy	93.0%	89.4%	87.8%

In summary, according to the evaluation results, the proposed P-CA scheme effectively achieves the edge–cloud interaction mechanism for HBR. Such a mechanism means the edge clients are able to employ complex deep learning models. However, in contrast to the federated learning approach, federated learning does not sufficiently consider the problem of limited computational power at the edge. As a result, although federated learning is capable of privacy-preserving edge–cloud interactions, it cannot leverage server computing power to support the deployment of complex models. In this work, the proposed P-CA is able to effectively address this problem.

4.2.3. Performance Comparison with the State-of-the-Art Methods

To quantify the level of privacy protection offered by our proposed mixture strategy, we proposed an evaluation metric named the privacy protection rate (PPR). The PPR is defined as $PPR = (ACC_{P-CA} - ACC_{TM})/ACC_{P-CA}$. Here, ACC_{TM} denotes the recognition accuracy achieved by the threat model, and ACC_{P-CA} denotes the accuracy obtained by the proposed P-CA. We assume the threat model as an attack scenario in which the cloud server has full access to the entire model, including the classifier. This scenario represents one of the most severe attack conditions as the cloud server normally lacks full access to the model from the edge. It implies that if the server obtains the raw data, it could infer real human behaviors. However, by employing the mixture strategies, even if the cloud server manages to steal the whole model it would only achieve poor recognition performance.

Table 3 summarizes the accuracy obtained by a threat model that uses mixed data or mixed features as input and the corresponding PPR of the proposed P-CA under these two scenarios. The results show that the threat model (data mix) achieves an accuracy of 37.4% and the threat model (feature mix) achieves an accuracy of 42.9%. Both accuracies are far below 50%, indicating failing recognition. The corresponding PPR values of the proposed P-CA are 0.60 and 0.54, respectively. Given the severity of the assumed threat model, it can be inferred that P-CA with mixture strategies could achieve even better PPR under less severe attack scenarios.

Table 3. Privacy protection rate (PPR) of the proposed P-CA under two threat models, including threat model (data mix) and threat model (feature mix).

Dataset	**Threat Model (data mix)**	**Threat Model (feature mix)**	**PPR (data mix)**	**PPR (feature mix)**
L-dataset	37.4%	42.9%	0.60	0.54

To further demonstrate the performance of the proposed P-CA, we conducted comparative experiments using two representative state-of-the-art technologies: FedLoc [24] and PPDFL [25]. FedLoc is designed based on the mainstream federated averaging (FedAvg) mechanism, which is a benchmark method for human behavior recognition. PPDFL is a state-of-the-art method enhanced by the theory of convex-hull optimization for human behavior recognition. Both of the two methods protect client privacy at the edge by training a central global model without requiring clients to share their original data.

No matter whether FedLoc, PPDFL, or our proposed P-CA, they all aim to achieve high recognition accuracy while simultaneously protecting privacy. FedLoc and PPDFL preserve

privacy by sharing gradients instead of raw data. And our P-CA models protect privacy by mixing raw data or feature maps. Therefore, we compare the accuracy obtained by P-CA and the aforementioned two methods. Table 4 summarizes the localization accuracy achieved by FedLoc, PPDFL, and our proposed P-CA models. The results show that P-CA (feature mix) achieves the highest accuracy of 79.6%, outperforming PPDFL and FedLoc by 2.1% and 2.7%, respectively. P-CA (data mix) also surpasses FedLoc by 0.5%. These findings demonstrate that our proposed model can achieve superior recognition performance while ensuring privacy protection. Additionally, unlike FedLoc and PPDFL, which rely solely on edge devices to run client models, our P-CA leverages an edge–cloud interaction architecture. This allows it to utilize server computing power, enabling the deployment of more complex models and yielding more promising recognition performance.

Table 4. Localization accuracy comparison of the proposed P-CA with Federated Localization algorithm (FedLoc) [24] and Privacy-Preserving Device-Free Localization algorithm (PPDFL) [25].

Dataset	FedLoc [24]	PPDFL [25]	P-CA (data mix)	P-CA (feature mix)
L-dataset	76.9%	77.5%	77.4%	79.6%

5. Discussion and Conclusions

In this work, we propose a trustworthy edge–cloud collaborative deep learning model interaction method, known as P-CA. Utilizing strategies such as network structure splitting and data mixing, P-CA allows the edge to effectively utilize the high computational resources of cloud servers to deploy complex models for precise human behavior recognition. Simultaneously, it significantly reduces the risk of privacy leakage of users' data.

Extensive experimental evaluations are conducted under various conditions on both our testbed localization dataset and publicly available activity recognition datasets. Experimental results show that our approach preserves the correct distribution of data and underlying patterns. The P-CA with data mixing achieves a comparable activity recognition accuracy of 89.4%, while the P-CA with feature mixing obtains a location recognition accuracy of 88.0%, which outperforms the state-of-the-art methods FedLoc and PPDFL. The corresponding PPRs of the proposed P-CA are 0.60 and 0.54. These observations demonstrate that the proposed P-CA effectively preserves privacy, reduces edge computations, and maintains acceptable HBR accuracy.

Discussion: There are three promising directions for future work based on the current research. First, due to the potential noise and variability in the collected CSI data, developing robust noise-resistant methods is crucial for the field of HBR. Models such as variational autoencoders and sparse autoencoders could effectively mitigate noise interference. Second, addressing the trade-off between privacy preservation and accuracy is essential. In this study, mixing coefficients were manually adjusted through trial-and-error experiments, which constrained the search range. Future research should explore more effective methodologies, such as heuristic algorithms like particle swarm optimization, to determine multiple weight factor distributions or reinforcement learning for multi-objective optimization. Finally, investigating transfer learning and domain adaptation methods could enhance the model's robustness and generalization ability across varying targets and scenarios.

Author Contributions: H.W. performed formal analysis, methodology, original draft, and review; C.Q. performed investigation; C.Z. performed the data curation; J.X. performed visualization; C.S. performed supervision, draft review, and funding acquisition. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported by JSPS Grant-in-Aid for Scientific Research (C) 23K11103.

Data Availability Statement: The datasets presented in this article are not readily available because the data are part of an ongoing study.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Huang, H.; Lin, L.; Zhao, L.; Huang, H.; Ding, S. TSHNN: Temporal-Spatial Hybrid Neural Network for Cognitive Wireless Human Activity Recognition. *IEEE Trans. Cogn. Commun. Netw.* 2024, *early access*.
2. Zhao, L.; Huang, H.; Wang, W.; Zheng, Z. An accurate approach of device-free localization with attention empowered residual network. *Appl. Soft Comput.* **2023**, *137*, 110164. [CrossRef]
3. Zhu, X.; Qu, W.; Qiu, T.; Zhao, L.; Atiquzzaman, M.; Wu, D.O. Indoor intelligent fingerprint-based localization: Principles, approaches and challenges. *IEEE Commun. Surv. Tutor.* **2020**, *22*, 2634–2657. [CrossRef]
4. Roy, P.; Chowdhury, C. A survey on ubiquitous WiFi-based indoor localization system for smartphone users from implementation perspectives. *CCF Trans. Pervasive Comput. Interact.* **2022**, *4*, 298–318. [CrossRef]
5. Hillyard, P.; Patwari, N. Never use labels: Signal strength-based bayesian device-free localization in changing environments. *IEEE Trans. Mob. Comput.* **2019**, *19*, 894–906. [CrossRef]
6. Wang, P.; Huang, H.; Zhao, L.; Zhu, B.; Huang, H.; Wu, H. ExtRe: Extended Temporal-Spatial Network for Consumer-Electronic WiFi-based Human Activity Recognition. *IEEE Trans. Consum. Electron.* 2024, *early access*. [CrossRef]
7. Wu, D.; Zeng, Y.; Gao, R.; Li, S.; Li, Y.; Shah, R.C.; Lu, H.; Zhang, D. WiTraj: Robust Indoor Motion Tracking with WiFi Signals. *IEEE Trans. Mob. Comput.* **2021**, *22*, 3062–3078. [CrossRef]
8. Wu, D.; Zeng, Y.; Zhang, F.; Zhang, D. WiFi CSI-based device-free sensing: From Fresnel zone model to CSI-ratio model. *CCF Trans. Pervasive Comput. Interact.* **2021**, *4*, 88–102. [CrossRef]
9. Shit, R.C.; Sharma, S.; Puthal, D.; James, P.; Pradhan, B.; van Moorsel, A.; Zomaya, A.Y.; Ranjan, R. Ubiquitous localization (UbiLoc): A survey and taxonomy on device free localization for smart world. *IEEE Commun. Surv. Tutor.* **2019**, *21*, 3532–3564. [CrossRef]
10. Zhao, L.; Yang, Q.; Huang, H.; Guo, L.; Jiang, S. Intelligent wireless sensing driven metaverse: A survey. *Comput. Commun.* **2024**, *214*, 46–56. [CrossRef]
11. Zhang, Y.; Liu, Q.; Wang, Y.; Yu, G. CSI-based location-independent human activity recognition using feature fusion. *IEEE Trans. Instrum. Meas.* **2022**, *71*, 1–12. [CrossRef]
12. Lin, S.; Xu, Y.; Wang, H.; Gu, J.; Liu, J.; Ding, G. Secure Multicast Communications via RIS Against Eavesdropping and Jamming with Imperfect CSI. *IEEE Trans. Veh. Technol.* **2023**, *71*, 5503312. [CrossRef]
13. He, Z.; Zhang, X.; Wang, Y.; Lin, Y.; Gui, G.; Gacanin, H. A Robust CSI-based Wi-Fi Passive Sensing Method Using Attention Mechanism Deep Learning. *IEEE Internet Things J.* **2023**, *10*, 17490–17499. [CrossRef]
14. Moghaddam, M.G.; Shirehjini, A.A.N.; Shirmohammadi, S. A WiFi-based Method for Recognizing Fine-grained Multiple-Subject Human Activities. *IEEE Trans. Instrum. Meas.* **2023**, *72*, 2520313. [CrossRef]
15. Zhang, L.; Ding, E.; Hu, Y.; Liu, Y. A novel CSI-based fingerprinting for localization with a single AP. *EURASIP J. Wirel. Commun. Netw.* **2019**, *2019*, 51. [CrossRef]
16. Li, H.; Chen, X.; Wang, J.; Wu, D.; Liu, X. DAFI: WiFi-based Device-free Indoor Localization via Domain Adaptation. *Proc. ACM Interact. Mob. Wearable Ubiquitous Technol.* **2021**, *5*, 1–21. [CrossRef]
17. Zhang, X.; He, F.; Chen, Q.; Jiang, X.; Bao, J.; Ren, T.; Du, X. A differentially private indoor localization scheme with fusion of WiFi and bluetooth fingerprints in edge computing. *Neural Comput. Appl.* **2022**, *34*, 4111–4132. [CrossRef]
18. Parmar, D.; Rao, U.P. Dummy generation-based privacy preservation for location-based services. In Proceedings of the 21st International Conference on Distributed Computing and Networking, Kolkata, India, 4–7 January 2020; p. 1.
19. Li, Z.; Ma, J.; Miao, Y.; Wang, X.; Li, J.; Xu, C. Enabling Efficient Privacy-Preserving Spatio-Temporal Location-Based Services for Smart Cities. *IEEE Internet Things J.* **2023**, *11*, 5288–5300. [CrossRef]
20. Shaham, S.; Ding, M.; Liu, B.; Dang, S.; Lin, Z.; Li, J. Privacy Preservation in Location-Based Services: A Novel Metric and Attack Model. *IEEE Trans. Mob. Comput.* **2021**, *20*, 3006–3019. [CrossRef]
21. Shaham, S.; Ding, M.; Liu, B.; Dang, S.; Lin, Z.; Li, J. Privacy preserving location data publishing: A machine learning approach. *IEEE Trans. Knowl. Data Eng.* **2021**, *33*, 3270–3283. [CrossRef]
22. Iwasawa, Y.; Nakayama, K.; Yairi, I.; Matsuo, Y. Privacy Issues Regarding the Application of DNNs to Activity-Recognition using Wearables and Its Countermeasures by Use of Adversarial Training. In Proceedings of the IJCAI, Melbourne, Australia, 19–25 August 2017; pp. 1930–1936.
23. Liu, Z.; Wu, Z.; Gan, C.; Zhu, L.; Han, S. Datamix: Efficient privacy-preserving edge-cloud inference. In Proceedings of the European Conference on Computer Vision, Glasgow, UK, 23–28 August 2020; Springer: Berlin/Heidelberg, Germany, 2020; pp. 578–595.
24. Yin, F.; Lin, Z.; Kong, Q.; Xu, Y.; Li, D.; Theodoridis, S.; Cui, S.R. FedLoc: Federated learning framework for data-driven cooperative localization and location data processing. *IEEE Open J. Signal Process.* **2020**, *1*, 187–215. [CrossRef]
25. Huang, H.; Huang, T.; Wang, W.; Zhao, L.; Wang, H.; Wu, H. Federated Learning and Convex Hull Enhancement for Privacy Preserving WiFi-Based Device-Free Localization. *IEEE Trans. Consum. Electron.* **2023**, *70*, 2577–2585. [CrossRef]

26. Zhang, H.; Cisse, M.; Dauphin, Y.N.; Lopez-Paz, D. mixup: Beyond empirical risk minimization. In Proceedings of the International Conference on Learning Representations (ICLR2018), Vancouver, BC, Canada, 30 April–3 May 2018; pp. 1–13.
27. Liu, J.; Gu, Y.; Kamijo, S. Customer behavior recognition in retail store from surveillance camera. In Proceedings of the 2015 IEEE International Symposium on Multimedia (ISM), Miami, FL, USA, 14-16 December 2015; IEEE: New York City, NY, USA, 2015; pp. 154–159.
28. Tom, R.J.; Sankaranarayanan, S.; Rodrigues, J.J. Smart energy management and demand reduction by consumers and utilities in an IoT-fog-based power distribution system. *IEEE Internet Things J.* **2019**, *6*, 7386–7394. [CrossRef]
29. Li, Q.; Liao, X.; Liu, M.; Valaee, S. Indoor localization based on CSI fingerprint by siamese convolution neural network. *IEEE Trans. Veh. Technol.* **2021**, *70*, 12168–12173. [CrossRef]
30. Zhang, B.; Sifaou, H.; Li, G.Y. CSI-fingerprinting indoor localization via attention-augmented residual convolutional neural network. *IEEE Trans. Wirel. Commun.* **2023**, *22*, 5583–5597. [CrossRef]
31. Wang, D.; Yang, J.; Cui, W.; Xie, L.; Sun, S. Multimodal CSI-based human activity recognition using GANs. *IEEE Internet Things J.* **2021**, *8*, 17345–17355. [CrossRef]
32. Yang, J.; Zhou, Y.; Huang, H.; Zou, H.; Xie, L. MetaFi: Device-free pose estimation via commodity WiFi for metaverse avatar simulation. In Proceedings of the 2022 IEEE 8th World Forum on Internet of Things (WF-IoT), Yokohama, Japan, 26 October–11 November 2022; IEEE: New York City, NY, USA, 2022; pp. 1–6.
33. Zhou, Y.; Huang, H.; Yuan, S.; Zou, H.; Xie, L.; Yang, J. MetaFi++: WiFi-enabled Transformer-based Human Pose Estimation for Metaverse Avatar Simulation. *IEEE Internet Things J.* **2023**, *10*, 14128–14136. [CrossRef]
34. Chang, Z.; Liu, S.; Xiong, X.; Cai, Z.; Tu, G. A survey of recent advances in edge-computing-powered artificial intelligence of things. *IEEE Internet Things J.* **2021**, *8*, 13849–13875. [CrossRef]
35. Gumaei, A.; Al-Rakhami, M.; AlSalman, H.; Rahman, S.M.M.; Alamri, A. DL-HAR: Deep learning-based human activity recognition framework for edge computing. *Comput. Mater. Contin.* **2020**, *65*, 1033–1057. [CrossRef]
36. Kwon, H.; Hedge, C.; Kiarashi, Y.; Madala, V.S.K.; Singh, R.; Nakum, A.; Tweedy, R.; Tonetto, L.M.; Zimring, C.M.; Clifford, G.D. Indoor Localization and Multi-person Tracking Using Privacy Preserving Distributed Camera Network with Edge Computing. *arXiv* **2023**, arXiv:2305.05062.
37. Wu, Q.; He, K.; Chen, X. Personalized federated learning for intelligent IoT applications: A cloud-edge based framework. *IEEE Open J. Comput. Soc.* **2020**, *1*, 35–44. [CrossRef] [PubMed]
38. Narayana, S.; Rao, V.; Prasad, R.V.; Kanthila, A.K.; Managundi, K.; Mottola, L.; Prabhakar, T.V. LOCI: Privacy-aware, device-free, low-power localization of multiple persons using IR sensors. In Proceedings of the 2020 19th ACM/IEEE International Conference on Information Processing in Sensor Networks (IPSN), Sydney, NSW, Australia, 21–24 April 2020; IEEE: New York City, NY, USA, 2020; pp. 121–132.
39. Cominelli, M.; Kosterhon, F.; Gringoli, F.; Cigno, R.L.; Asadi, A. An experimental study of CSI management to preserve location privacy. In Proceedings of the 14th International Workshop on Wireless Network Testbeds, Experimental Evaluation & Characterization, London, UK, 21 September 2020; pp. 64–71.
40. Shi, Q.; Zhang, Z.; He, T.; Sun, Z.; Wang, B.; Feng, Y.; Shan, X.; Salam, B.; Lee, C. Deep learning enabled smart mats as a scalable floor monitoring system. *Nat. Commun.* **2020**, *11*, 4609. [CrossRef]
41. Halperin, D.; Hu, W.; Sheth, A.; Wetherall, D. Tool release: Gathering 802.11 n traces with channel state information. *ACM SIGCOMM Comput. Commun. Rev.* **2011**, *41*, 53. [CrossRef]
42. Hao, Y.; Wang, W.; Lin, Q. Incident retrieval and recognition in video stream using wi-fi signal. *IEEE Access* **2021**, *9*, 100208–100222. [CrossRef]
43. Zhao, L.; Huang, H.; Li, X.; Ding, S.; Zhao, H.; Han, Z. An accurate and robust approach of device-free localization with convolutional autoencoder. *IEEE Internet Things J.* **2019**, *6*, 5825–5840. [CrossRef]
44. Yousefi, S.; Narui, H.; Dayal, S.; Ermon, S.; Valaee, S. A survey on behavior recognition using WiFi channel state information. *IEEE Commun. Mag.* **2017**, *55*, 98–104. [CrossRef]