A differentially private indoor localization scheme with fusion of WiFi and bluetooth fingerprints in edge computing

Xuejun Zhang¹ · Fucun He¹ · Qian Chen^{1,2} · Xinlong Jiang² · Junda Bao¹ · Tongwei Ren³ · Xiaogang Du¹

Received: 13 January 2021 / Accepted: 25 November 2021 / Published online: 28 January 2022
© Crown 2021

Abstract

As an enabling technology for edge computing scenarios, indoor localization has a broad prospect in a variety of location-based applications, such as tracking, navigating, and monitoring in indoor environments. In order to improve the location accuracy, numerous machine learning (ML)-based indoor localization schemes with fingerprint fusion have been proposed recently, which take advantage of the fusion of signal gathered from multiple wireless technologies (e.g., WiFi and BLE) and require a site survey to construct the fingerprint database. However, most solutions are based on cloud framework and thus pose a serious privacy leakage because users' sensitive information (e.g., locations) is computed from the fingerprint database by the untrusted localization service provider. Furthermore, the site survey is time-consuming and labor-intensive. In this paper, we propose a differentially private fingerprint fusion semi-supervised extreme learning machine for indoor localization in the edge computing, called Adp-FSELM. The Adp-FSELM firstly employs a multi-level edge network-based privacy-preserving system framework to meet the requirements of ML-based fingerprint indoor localization for lightweight, low latency, and real-time response. Then, the Adp-FSELM extends the ϵ -differential privacy to the fingerprint fusion semi-supervised extreme learning machine for indoor localization in edge computing through a three-phase private process consisting of private labeled sample obfuscation, differentially private feature fusion, and differentially private model training. Theoretical and comprehensive experimental results in real indoor environments demonstrate that the Adp-FSELM provides a high ϵ -differential privacy guarantee for users' location privacy while reducing human calibration effort and effectively resists Bayesian inference attacks. Compared with the existing semi-supervised learning-based localization methods, the mean absolute error of location accuracy of the Adp-FSELM is restricted to 2.22% at most, and the additional time consumption can be almost ignored. Thus, our mechanism can balance the trade-off among location privacy, location accuracy, and time consumption.

Keywords Location privacy · Edge computing · Differential privacy · Fusion semi-supervised extreme learning machine · Indoor localization

1 Introduction

The proliferation of Internet of everything (IoE) and sensor-rich smart IoE devices has spawned a wide range of indoor location-based services (ILS), such as indoor guidance and navigation, smart inventory, context-aware location-based marketing, personnel care and elderly care, disaster management and assist, and so forth. However, satellite-based positioning services are not sufficient to meet the localization (positioning) requirements of such

indoor location applications, mainly because the GPS signals are almost not available in an indoor environment. As a key technology in ILS, indoor positioning has become very hot research field in recent years [1] and has also become a very important direction of the development of the IoE and smart city [2]. Particularly, fingerprint-based indoor localization has recently attracted extensive attention and interest both from the industry and from the academic due to its promising performance (e.g., easy measurement, easy implementation, and low cost) [3]. Fingerprint-based indoor localization usually takes advantage of the received signal strength indicator (RSSI) from

Extended author information available on the last page of the article

multiple access points (APs) to estimate users' locations [4, 5], which relies on cloud-based indoor positioning system (IPS) combined with user terminal and cloud positioning server (CPS) and is supported by lots of commercial companies such as Wi-FiSLAM, Google, Navizon, and so on [6]. The core of fingerprint-based indoor positioning is localization algorithm that can find the mapping relationship between fingerprints and location [7].

At present, machine learning (ML) is an effective tool to construct this mapping [2], which has been widely used in fingerprint-based indoor localization [8]. The ML-based fingerprint indoor localization mainly consists of offline training stage and online positioning stage. In the offline training stage, the CPS measures the RSSI from various APs (e.g., Wi-Fi and BLE) which are deployed in various predefined locations and records these RSSI fingerprints and their corresponding locations in a fingerprint database for a statistical learning model. Subsequently, in the online positioning stage, user sends his/her current RSSI at a certain location to the statistical learning model on the CPS, in which the localization algorithm searches the fingerprint database to find the fingerprint that best matches the submitted by user in real-time RSSI to estimate user's accurate location.

In order to obtain high-precision positioning, the ML-based fingerprint IPS requires a big amount of labeled RSSI data in training phase. However, RSSI data calibration is time-consuming and strenuous. Therefore, how to obtain the high positioning accuracy with less manual calibration is a main challenge for the ML-based fingerprint indoor positioning. Recently, fingerprint crowdsourcing [7] and FSELM [8] have gained attention to reduce the difficulty of data calibration while ensuring location accuracy by aggregating or fusing small data collected from a large number of mobile users. Fingerprint crowdsourcing is an essential method for constructing large fingerprint datasets effectively for ML-based fingerprint indoor localization training, which allows mobile users to contribute their measurements in a participatory sensing manner [7]. The FSELM is a novel ML-based fingerprint indoor positioning method, which utilizes the fused RSSI fingerprints of Wi-Fi and BLE to achieve enhanced location accuracy, with little manual calibration effort [8]. By adopting the manifold regularization technology and using both labeled and unlabeled RSSI fingerprints, the FSELM addresses the problem that most existing ML-based indoor localization methods require large number of labeled RSSI fingerprints for model training. Although fingerprint crowdsourcing and FSELM can effectively address the above challenge of ML-based indoor fingerprint localization model, they jeopardize user's location privacy due to their adoption of cloud-based IPS. What's more, the ML-based fingerprint IPS is a typical low-latency, high-

precision, and compute-intensive service. It has the characteristics of more frequent user localization requests, rapid deployment of indoor localization environments, more complex data fusion of joint localization in IoE environment, and a large quantity of data collection, transmission, and computation to improve location accuracy. All of these also pose several challenges to the cloud-based IPS. First, the network bandwidth and reliability for data transmission between the user terminal and CPS cannot be guaranteed due to the complexity of the wireless network environment. At the same time, long-distance data transportation from the user terminal to CPS will lead to huge unnecessary network delay. Second, user's localization data are generated at the edge of the network and involve a lot of personal privacy. If all the localization data need to be sent to the CPS, the privacy breaches will pose an obstacle for cloud-based IPS. Third, most of the user terminal devices in cloud-based IPS have limited battery capacity, which causes a critical energy consumption constraints. Meanwhile, the operation and maintenance of CPS also require a large amount of energy due to centralized processing of localization services. Therefore, offloading some localization calculation tasks to the edge nodes may be more energy efficient. Lastly, cloud-based IPS cannot meet the requirements of the system boundary area for fast real-time localization task processing.

To address these challenges, one possible solution is to adopt edge-cloud collaborative system to deploy ML-based IPS. The nature characteristics of edge computing such as low latency, location awareness, distribution, programmability, and data localization just meet the requirements of big data processing, positioning delay, and accuracy improvement brought by cloud-based fingerprint IPS that utilizes ML algorithms. However, in edge-cloud collaborative system, location privacy concern of ML-based fingerprint IPS is still severe due to the lack of effective data privacy-preserving mechanisms for edge computing infrastructure with resource constrained. During offline training and online positioning of indoor localization, it is still possible for untrusted CPS and edge server to derive user's accurate location by analyzing the labeled RSSI data with abundant feature information from edge devices, and thereby locating or tracking user within a small area near the Wi-Fi router [9]. What's more, the loss of location privacy may cause user's social reputation damage, economic damage, or even physical violence [5].

Most of the existing indoor localization privacy-preserving mechanisms, e.g., k -anonymity-based methods [6, 10], encryption-based approaches [5, 11, 12], and differential privacy (DP)-based methods [13–15], mainly focus on protecting the privacy breach in online positioning phase of cloud-base IPS against untrusted CPS. Nevertheless, these privacy-preserving mechanisms are not fully

applicable to edge-based IPS, especially in offline training phase, because they are resource hungry. In addition, few studies have paid much attention to the privacy issues of fingerprint fusion indoor localization machine learning model training.

In this paper, we introduce an edge–cloud collaboration privacy-preserving IPS with fusion of WiFi and BLE fingerprint and propose A differentially private fingerprint Fusion Semi-supervised Extreme Learning Machine, named Adp-FSELM, a novel and effective privacy-preserving indoor localization ML model. *Firstly*, we design a four-level edge–cloud collaboration network to meet the needs of ML-based fingerprint IPS for lightweight, low latency, real-time response, fast training, and high precision. *Secondly*, we add appropriate Laplace noise into each operation step of the Adp-FSELM to prevent privacy disclosure from the users' labeled RSSI training samples and utilize ϵ -differential privacy technique to guarantee its privacy and security. *Finally*, we employ a small set of labeled RSSI samples as well as a large number of unlabeled RSSI samples fused with the WiFi and BLE fingerprints to improve the positioning performance of the Adp-FSELM with little manual calibration effort. In this way, comprehensive experimental results demonstrate that, compared to five semi-supervised learning methods on location accuracy and six non-private localization methods on time consumption, the Adp-FSELM can provide the provably privacy-preserving with little resource cost and performance sacrifice.

Our contributions can be summarized as follows:

- (1) To the best of our knowledge, the Adp-FSELM is the first work to address the privacy-preserving for ML-based fingerprint fusion IPS in edge computing. By conducting the trusted aggregation and fusion model of multiple RSSI fingerprints, the Adp-FSELM succeeds in utilizing the distributed edge computing resources and differential privacy technique to obtain the accurate location accuracy without leaking user's location privacy.
- (2) We propose a differentially private fingerprint fusion indoor positioning ML model in edge–cloud collaboration system. Besides, we provide multi-level privacy budget allocation strategy to satisfy different privacy requirements as regard what needs to be preserved.
- (3) We establish comprehensive experiments on practical datasets. The results show that, compared with five semi-supervised learning methods on location accuracy and six mature localization methods on time consumption, the Adp-FSELM achieves the high accuracy with low time consumption while protecting location privacy.

The rest of the paper is organized as follows. Section 2 reviews the related work. The preliminaries of our method are provided in Sect. 3. In Sect. 4, we give our method in detail. The experiments and performance evaluation are shown in Sect. 5. Finally, we conclude the paper.

2 Related work

In this section, we review the advanced ML-based indoor localization technologies, edge computing, and the privacy-preserving indoor positioning technologies, respectively.

2.1 Indoor localization

Recently, ML methods have proved to be a practical approach for IPS [6]. However, there are still many challenges to be solved in the complex and dynamic indoor localization scenario. For example, indoor localization model needs an enormous amount of labeled RSSI training samples to generate a good location estimation model. Generally, collecting this large number of labeled RSSI data is very time-consuming and tedious. By contrast, it is easier and cheaper to obtain unlabeled RSSI data. Therefore, several attempts have been proposed to improve localization performance by using massive unlabeled RSSI data combined with a little labeled RSSI data.

In the existing ML-based indoor positioning localization methods, manifold learning and semi-supervised regularization methods have been widely studied and adopted because of their excellent properties and performance. Liu et al. [16] proposed the SELM method with graph Laplacian regularization, which can achieve good positioning performance and a fast learning speed for sparsely calibrated localization problems. To improve localization performance, Gu et al. [17] proposed SDELM approach for IPS. The SDELM not only takes full advantage of deep learning (DL), semi-supervised learning, and extreme learning machine (ELM) but also reduces the human calibration effort with the help of unlabeled RSSI data. Chen et al. [18] proposed an ML-based IPS in a wireless sensor network environment, which uses the non-anchor information to improve localization performance. Above all, these ML-based IPS achieve a high location accuracy by taking advantage of ML and semi-supervised learning methods as well as both labeled and unlabeled data.

As we all know, the performance of ML-based IPS depends not only on the amount of labeled RSSI training samples, but also on the diversity of RSSI training data. Consequently, some practical ML-based IPS have been explored to improve location accuracy by considering the fusion of multiple wireless signals (e.g., WiFi and BLE). In

[19], the author proposed a fusion method for indoor localization through merging Bluetooth and WLAN technologies. In [20, 21], based on trilateration techniques, the authors presented a propagation model and algorithm to obtain high location accuracy by combining BLE with WiFi. In [8], FSELM was introduced to improve the location accuracy of ML by fusing BLE and BLE signals. The experimental results show that the FSELM can achieve excellent performance while reducing the human calibration effort for RSSI training samples. In [22], the LeMan approach was presented to reduce calibration effort for sensor network-based tracking based on the manifold regularization. In [23], the random vector functional link (RVFL) method was introduced to obtain the enhanced location accuracy by considering both the links between enhanced pattern and outputs and the direct links from model inputs to outputs. In [8], the semi-supervised RVFL (S-RVFL) and the fusion semi-supervised RVFL (FS-RVFL) were constructed based on the RVFL for the performance comparison of indoor localization methods. Compared with the case of WiFi alone, these indoor positioning technologies fuse multiple RSSI signals to achieve higher positioning accuracy through fusion map with WiFi and BLE. However, the privacy issues of these methods have never been addressed in the studies.

2.2 Edge computing

With the proliferation of IoE and smart cities, the research on indoor positioning technology has become a hot spot [2]. The existing indoor positioning systems rely on cloud computing framework to determine the accurate location of user upon request [6]. However, cloud-based indoor localization systems trigger some problems, such as high network delay, privacy leaks, energy consumption, inability to meet the needs of fast and real-time positioning task processing, and so on. Edge computing [24] has the potential to address these problems. Compared with cloud computing, edge computing processes and stores data at the edge of the mobile network, which decreases the bandwidth load, reduces the response time, improves the battery life, ensures data security and privacy, and offers an improved user experience [25]. It just meets the requirements of low latency, real-time response, fast training, and high precision for indoor localization system. Satyanaraya et al. [26] proposed the cloudlet technology at the edge of Internet to deploy mobile cloud service. However, it was inadequate due to its limited WiFi coverage. Abbas et al. [25] proposed that mobile edge networks can make the network with low latency and high bandwidth because they are equipped with better offloading technologies. Hu et al. [27] pointed out that edge computing can greatly reduce the network delay and thus implement the computation-

intensive applications in WiFi networks. Du et al. [28] proposed an ML algorithm for edge computing and used DP technology to ensure the privacy and security of the training datasets in wireless big data scenarios.

2.3 Privacy-preserving indoor localization

Only a few works on privacy-preserving indoor localization have been proposed in the literature. Konstantinidis et al. [6] proposed a privacy-preserving indoor positioning approach based on k -anonymity that has been thoroughly studied in some areas (e.g., privacy-preserving medical data). This solution conceals the user's authentic location trace among $k - 1$ fake traces. In this way, the CPS is supposed not to use any supplementary information, including statistics (e.g., average numbers of users in concrete areas), to verify the users' requests against the building map. In fact, however, using such auxiliary information can allow the adversary to differentiate authentic traces, thereby tracing both past and coming movements of the user. Eshun et al. [11] put forward an indoor location privacy-preserving protocol, which can permit a CPS to acquire users' locations while not infringing their location privacy. Although the protocol protects both users' location privacy and the CPS's data privacy, it still provides good ILS. In this scheme, neither the CPS nor the client trusts each other, so it is a typical secure multi-party computation problem. Li et al. [5] built PriWFL using additive homomorphic Paillier encryption to protect both the CPS's data privacy and the user's location privacy. Although fully homomorphic encryption technology can return correct results without decrypting user's queries, its efficiency still needs to be further improved. Yang et al. [12] proposed two privacy-preserving schemes based on Paillier encryption to avoid privacy risks associated with the weakness of PriWFL while maintaining the same location accuracy. Wang et al. [29] introduced LoPEC based on noise addition in mobile edge computing environment, in which the signal coverage relationship between APs is simplified into an undirected graph. The three APs circles determine a location (e.g., a complete subgraph), and the nearby APs complete subgraph is used as a noise location fingerprint. However, in reality, it is difficult for the three circles to intersect at one point. Besides, to improve location accuracy, there are usually more than three APs participated in the calculation. Therefore, the effectiveness of noise locations is still a challenging issues. There are also some effective methods utilizing the differential privacy to mitigate the privacy concern for indoor localization systems. Wang et al. [13] proposed DP3 to preserve both clients' location privacy and CPS's data privacy in WiFi fingerprint-based IPS.

Zhao et al. [14] proposed P^3 -LOC to solve the privacy problem of both user's location and CPS's data privacy. The P^3 -LOC employs k -anonymity and DP technique to achieve provable privacy protection for a two-stage localization paradigm. In [15], inLocDiff was proposed to guarantee a certain level of privacy in WiFi fingerprint-based IPS, which obfuscates the RSSI data with controllable noise to trade off the privacy and utility. However, these schemes are designed for the online operating phase and are not directly applicable to the privacy-preserving for ML-based multiple fusion indoor localization training model with a more complex location estimation algorithm. Moreover, they did not consider some features of IPS in the edge computing scenarios, such as low latency, real-time response, and resource constraint.

Different from the existing schemes, to the best of our knowledge, the Adp-FSELM is the first effort to address the privacy issue for ML-based fingerprint fusion indoor localization model training in edge computing scenarios. It not only guarantees the users' location privacy but also obtains acceptable location accuracy with low resource consumption and fast training speed. Through rigorous mathematical derivation and privacy analysis, we prove that DP can achieve reliable privacy-preserving for RSSI training fingerprints of ML-based indoor localization model.

3 Preliminaries

In this section, we first presented our previous FSELM for indoor localization. Then, the edge-cloud collaboration system adopted in our method is explained. Finally, we introduce differential privacy preliminaries.

3.1 FSELM: fusion semi-supervised extreme learning machine

FSELM [8] is an extension of the extreme learning machine (ELM) [30], which fuses both WiFi and BLE signals into a unified model to improve location accuracy with a little manual calibration effort. It utilizes semi-supervised manifold regularization and graph Laplacian regularization to learn from sparsely calibrated RSSI samples for meeting the different propagation characteristics and effective distances of WiFi and BLE signals. Compared with other traditional IPS based on semi-supervised learning, our FSELM not only achieves better location accuracy, but also requires less training and prediction time [8]. Figure 1 shows the structure of FSELM with L hidden nodes.

Fig. 1 FSELM structure with L hidden nodes

In Fig. 1, we assume that there are N arbitrary distinct $n \times 1$ input signal feature vector $\mathbf{x}_i = [x_{i1}, x_{i2}, \dots, x_{in}]^T$ and $m \times 1$ target physical location $\mathbf{t}_i = [t_{i1}, t_{i2}, \dots, t_{im}]^T$, $i = 1, 2, \dots, N$, the output function of this network with hidden nodes is:

$$f_L(\mathbf{x}_j) = \sum_{i=1}^L \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}_j), \quad j = 1, \dots, N, \quad (1)$$

where $\mathbf{a}_i, b_i$ are two learning parameters of hidden layer nodes, $\mathbf{a}_i = [a_{i1}, a_{i2}, \dots, a_{in}]$ is the weight vector connecting the i th hidden layer node and input nodes, and b_i is the bias of the i th hidden layer node. L is the number of the hidden nodes, and β_i is the weight connecting the i th hidden layer node to output nodes. $G(\mathbf{a}_i, b_i, \mathbf{x}_j)$ is the output of the i th hidden node. If the activation function of the i th hidden layer node is $g(x) : \mathbf{R} \rightarrow \mathbf{R}$, then the $G(\mathbf{a}_i, b_i, \mathbf{x}_j)$ can be represented as formula (2):

$$G(\mathbf{a}_i, b_i, \mathbf{x}_j) = g(\mathbf{a}_i \cdot \mathbf{x}_j + b_i), \quad b_i \in \mathbf{R}. \quad (2)$$

If the FSELM network with L hidden nodes can approximate these N samples $(\mathbf{x}_i, \mathbf{t}_i)$ with zero error, then for some set of value $\beta_i, \mathbf{a}_i$, and b_i , the N samples satisfy that

$$\mathbf{t}_j = f_L(\mathbf{x}_j) = \sum_{i=1}^L \beta_i G(\mathbf{a}_i, b_i, \mathbf{x}_j), \quad j = 1, \dots, N. \quad (3)$$

For the sake of simplicity, we denoted Eq. (3) as:

$$\mathbf{T} = \mathbf{f} = \mathbf{H}\boldsymbol{\beta}, \quad (4)$$

where $\mathbf{H}(\mathbf{a}_1, \dots, \mathbf{a}_L, b_1, \dots, b_L, \mathbf{x}_1, \dots, \mathbf{x}_N)$

$$= \begin{bmatrix} G(\mathbf{a}_1, b_1, \mathbf{x}_1) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_1) \\ \vdots & \ddots & \vdots \\ G(\mathbf{a}_1, b_1, \mathbf{x}_N) & \cdots & G(\mathbf{a}_L, b_L, \mathbf{x}_N) \end{bmatrix}_{N \times L}, \quad (5)$$

$$\boldsymbol{\beta} = \begin{bmatrix} \beta_1^T \\ \vdots \\ \beta_L^T \end{bmatrix}_{L \times m} \quad \text{and} \quad \mathbf{T} = \begin{bmatrix} \mathbf{t}_1^T \\ \vdots \\ \mathbf{t}_L^T \end{bmatrix}_{N \times m}, \quad (6)$$

By solving Eq. (4), we can obtain its smallest norm least-squares solution:

$$\hat{\boldsymbol{\beta}} = \mathbf{H}^\dagger \mathbf{T}, \quad (7)$$

where $\mathbf{H}^\dagger = (\mathbf{H}^T \mathbf{H})^{-1} \mathbf{H}^T$ is the Moore–Penrose generalized inverse of the output matrix $\mathbf{H}$.

In order to fuse WiFi and BLE signal into a unified model, the FSELM uses graph Laplacians to establish the manifold regularization for them, respectively. According to the literature [8], the graph Laplacians $\mathbf{L}$ is denoted as $\mathbf{L} = \mathbf{D} - \mathbf{W}$, where $\mathbf{W} = [W_{ij}]_{(l+u) \times (l+u)}$ is a weight matrix, and $w_{ij} = \exp(-||s_i - s_j||^2 / 2\sigma^2)$ if s_i and s_j are the nearest neighbors on the manifold constraint, otherwise $w_{ij} = 0$; $\mathbf{D} = \sum_{j=1}^{l+u} \mathbf{W}_{ij}$ is a diagonal matrix. In [8], the goal of the FSELM is expressed as

$$\arg_f \min \left\{ \frac{1}{2} \|\mathbf{f} - \mathbf{T}\|^2 + \lambda_1 \mathbf{f}^T \mathbf{L}_1 \mathbf{f} + \lambda_2 \mathbf{f}^T \mathbf{L}_2 \mathbf{f} \right\}, \quad (8)$$

where $\frac{1}{2} \|\mathbf{f} - \mathbf{T}\|^2$ is the empirical error in regard to the labeled training samples, $\lambda_1 \mathbf{f}^T \mathbf{L}_1 \mathbf{f}$ is the manifold constraint for WiFi based on the graph Laplacian $\mathbf{L}_1$ and coefficients λ_1 and $\lambda_2 \mathbf{f}^T \mathbf{L}_2 \mathbf{f}$ is the manifold constraint for BLE based on the graph Laplacian $\mathbf{L}_2$ and coefficient λ_2 . By Substituting Eq. (4) into formula (8), we can obtain:

$$\arg_{\boldsymbol{\beta}} \min l(\boldsymbol{\beta}) = \arg_{\boldsymbol{\beta}} \min \left\{ \frac{1}{2} \|\mathbf{J}\mathbf{H}\boldsymbol{\beta} - \mathbf{T}\|^2 + \lambda_1 (\mathbf{H}\boldsymbol{\beta})^T \mathbf{L}_1 \mathbf{H}\boldsymbol{\beta} + \lambda_2 (\mathbf{H}\boldsymbol{\beta})^T \mathbf{L}_2 \mathbf{H}\boldsymbol{\beta} \right\}, \quad (9)$$

where $\mathbf{J}$ is an indication matrix, $\mathbf{J} = \text{diag}(\delta_1; \delta_2; \dots; \delta_{l+u})$, if the i th sample is labeled $\delta_i = 1$ otherwise $\delta_i = 0$.

By solving the optimization problem expressed in formula (9), we can get

$$\frac{\partial l}{\partial \boldsymbol{\beta}} = 0 \Rightarrow (\mathbf{J}\mathbf{H}\boldsymbol{\beta} - \mathbf{T})^T \mathbf{J}\mathbf{H} + \lambda_1 (\mathbf{H}\boldsymbol{\beta})^T \mathbf{L}_1 \mathbf{H} + \lambda_2 (\mathbf{H}\boldsymbol{\beta})^T \mathbf{L}_2 \mathbf{H} = 0. \quad (10)$$

Thus,

$$\boldsymbol{\beta} = ((\mathbf{J} + \lambda_1 \mathbf{L}_1^T + \lambda_2 \mathbf{L}_2^T) \mathbf{H})^\dagger \mathbf{J} \mathbf{T}. \quad (11)$$

Finally, the FSELM uses the model $\mathbf{f} = \mathbf{H}\boldsymbol{\beta}$ to generate the location prediction.

Obviously, the FSELM has a simple network structure and can randomly generate all learning parameters of hidden nodes based on any continuous sampling distribution. It not only has fast learning speed but also markedly improves the location accuracy while reducing human calibration effort. However, the FSELM is based on a cloud architecture. Thus, it may lead to high data transportation latency and serious location privacy disclosure. Furthermore, using fused RSSI sample from both WiFi and BLE for FSELM can increase the amount of feature information obtained, such that the probability of compromising user's location privacy is increased. Motivated by this, we propose the Adp-FSELM, a novel edge–cloud collaboration privacy-preserving IPS that use ML and DP.

3.2 Edge–cloud collaboration system

Figure 2 shows the collaboration edge–cloud architecture, which consists of three layers: the cloud server, edge servers, and edge devices [25]. The edge server resides between the cloud and edge servers and is regarded as the bridge between cloud server and edge devices interactions. It mostly collaborates with cloud server to enhance the performance of edge devices. Compared with cloud-only, the edge–cloud collaboration networks has three main strengths [31]: (1) backbone network alleviation, (2) agile service response, and (3) powerful cloud backup. Properly deploying deep learning or ML-based applications into edge–cloud collaboration system can achieve lower latency and power consumption than that into a cloud only or an edge-only system [32].

In this paper, we concentrate our research on privacy protection of the ML-based multi-fingerprints fusion indoor

Fig. 2 Collaboration edge–cloud architecture

localization model training, such as FSELM, in edge computing scenarios. We do not assume that edge computing may employ some specific technologies, as our solution is designed based on the rationales of the most of these technologies. The other technologies related to edge computing, such as energy management, data transmission, computing resource optimization, and operation efficiency, are out of our research scope, and interested reader can refer to the literature [26, 33]. As can be seen from the above introduction, the FSELM algorithm contains three major operation phases: (1) WiFi and BLE signal collection and aggregation, (2) WiFi and BLE signal feature fusion, and (3) localization model training. In the edge–cloud collaboration system, in order to obtain the better performance and privacy protection for FSELM, one possible solution is to deploy its three major operations on edge devices, edge servers, and cloud server separately. However, there can be privacy breaches of RSSI data during the FSELM localization training. To prevent the privacy leakage of the RSSI data, we add appropriate noise into each operation phase of the FSELM via DP technology. Besides, in order to ensure high positioning accuracy and good generalization capacity of the FSELM, the diversity of the RSSI data is indispensable. Therefore, we divide the edge devices layer into edge devices and edge nodes for obtaining the diverse RSSI training data (cf. Fig. 3).

3.3 Differential privacy

Differential privacy was firstly introduced by Dwork for addressing the privacy disclosure of statistical databases. It provides provable privacy guarantee for individuals in database when publishing statistical information and is supported by solid mathematical theories.

Definition 1 (ϵ -DP) Suppose there is a random algorithm $\mathcal{M}$ and $\mathcal{R}$ is all possible output sets of $\mathcal{M}$. For any two adjacent dataset D and D' (there is at most one different record between them), and any subset $\mathcal{S}$ of $\mathcal{R}$, $\mathcal{M}$ is ϵ -DP if $\mathcal{M}$ satisfies:

$$P[\mathcal{M}(D) \in \mathcal{S}] \leq e^\epsilon \cdot P[\mathcal{M}(D') \in \mathcal{S}], \quad (12)$$

where ϵ is the privacy budget and represents the privacy-preserving level that random algorithm $\mathcal{M}$ can provide. $P[\cdot]$ is a probability distribution function. This definition ensures that the presence or absence of an individual in the database has little influence on the final output of the query.

Definition 2 (Global sensitivity) Given a query function $Q: D \rightarrow \mathbf{R}^d$, the D is a dataset, and the output is a d -dimensional real vector. For any D and its adjacent dataset D' , the global sensitivity Δf of Q can be defined as follows:

$$\Delta f = \max_{D, D'} \|Q(D) - Q(D')\|_1, \quad (13)$$

where $\|\cdot\|_1$ represents the l_1 norm distance. Δf determines the amount of noise added to the query results, which shows the maximum influence of deleting any record in the dataset on query results.

Fig. 3 Edge–cloud collaboration system model for Adp-FSELM

In our Adp-FSELM, Δf is computed by using the maximal difference between any RSSI features in the dataset.

Definition 3 (*Laplace mechanism*) Given a function $f: D \rightarrow \mathbf{R}^d$ over a dataset D and its sensitivity Δf , the random algorithm in Eq. (14) is ε -DP.

$$\mathcal{M}(D) = f(D) + \text{Laplace}(\Delta f/\varepsilon), \quad (14)$$

where $\text{Laplace}(\Delta f/\varepsilon)$ is a random noise that follows the Laplace distribution with scale parameter $\Delta f/\varepsilon$.

The Laplace mechanism is a fundamental implementation mechanism of ε -DP protection, which is suitable for the protection of numerical query results (i.e., real-valued).

4 Our proposed method

In this section, we describe the Adp-FSELM scheme in detail, which is inspired by our previous FSELM [8] and EC-DPELM [34]. We first introduce the system model and privacy threat model. Then, we propose the Adp-FSELM algorithm, which is a differentially private indoor localization algorithm with fusion of WiFi and BLE fingerprints in edge–cloud collaboration system. Finally, we give concrete privacy analysis and description of privacy parameters.

4.1 System model

Figure 3 shows the edge–cloud collaboration architecture for Adp-FSELM. In this architecture, we divide the operation phases of Adp-FSELM across edge devices, edge nodes, edge servers, and cloud server. Assuming all the users voluntarily contribute their collected RSSI fingerprints to train the ML-based fingerprint fusion indoor localization model being hosted on cloud server, so as to enjoy the desired ILS. Meanwhile, users strive to protect their location privacy from being invaded by malicious entities (e.g., the untrusted cloud server, edge servers, and semi-trusted edge nodes). As shown in Fig. 3, the Adp-FSELM involves four entities, specifically as follows.

- *Edge devices* refer to a set of smart terminal devices (e.g., mobile phones, sensors) attached to users, which have computation, communication, and storage capabilities. In the offline training phase, the edge devices can be used to collect and store the WiFi and BLE signals from multiple wireless sensor beacons in an interested indoor area, execute the data partition and noise addition, and then transfer the processed RSSI data to nearby edge nodes. Meanwhile, in online operation phase, the edge devices send their measured

real-time RSSI fingerprints to edge servers to obtain localization services.

- *Edge nodes* are some smart gateways with the capability of data computation and storage, which can be regarded as part of edge devices, and may be deployed at a room, or an office, or a floor. They implement feature aggregation operation for different RSSI data (e.g., WiFi and BLE signals) from edge devices and transfer the aggregated RSSI data to the edge servers. The purpose of setting up edge nodes is twofold, one is to aggregate the rich RSSI data for feature fusion to reduce network delay and the other is to provide diversified fingerprint features for model training/testing to ensure its performance.
- *Edge servers* are core entities for edge–cloud collaboration architecture, which are implemented locally at user premises (e.g., parks, mall, and shopping center) and may be deployed at fixed location (e.g., base station) [25]. Compared with the edge nodes, the edge servers have more storage and computation resources. They execute the differential private fingerprint fusion operations for the FSELM and send the fused results to the cloud server. In the online state, the edge servers perform the indoor positioning model downloaded from the cloud server to provide the efficient indoor localization services for users.
- *Cloud server* has a strong capacity of data computation and storage, which is usually stored in a large-scale data center. It processes the extreme volume of the aggregation data for training ML-based indoor localization models by means of differential private and offloads the trained indoor localization model to the edge servers.

4.2 Threat model

We consider an edge–cloud collaboration indoor localization system, as shown in Fig. 3. The edge devices are responsible for collecting and storing diverse RSSI data (e.g., WiFi and BLE signals), and the edge nodes are used to aggregate these diverse RSSI data from different edge devices and also provide diversified aggregated RSSI for edge servers. The edge servers are employed to realize the fusing operation of WiFi and BLE signals, and the cloud server provides the powerful support for indoor positioning model training. In this paper, we mainly concentrate on location privacy disclosure of ML-based indoor localization model training in edge–cloud collaboration system. From the users' perspective, the attacker could be the cloud server, edge service, and edge node who need to use the RSSI fingerprints from the users during localization model training, or an external attacker who compromises the locations of smart terminal users and sells them for

benefits. Therefore, we consider that the edge server and the cloud server are untrusted, and both of them may attempt to steal participating users' location privacy derived from the RSSI fingerprints in the process of indoor localization model training. Specifically, the edge server and cloud server try to identify each user's location information containing in the RSSI fingerprints with the help of the positioning algorithm, and then further utilize these pieces of location information to analyze the pattern of users' trajectory, shopping habits, and salary level or even to conduct unauthorized data transactions.

Besides, the external attacker can maliciously analyze the difference of the localization model or some intermediate results in the training process, and speculate whether the attack target exists in the training samples or not, thus exposing the to-be-localized user's location privacy.

We also consider edge devices are trusted, that is, they correctly process the collected RSSI fingerprints and do not leak the privacy of these RSSI fingerprints to other parties. However, the edge nodes are considered semi-trusted, that is, they are curious about the collected RSSI fingerprints from users but will honestly aggregate these fingerprints. We assume they cannot collude with each other. Furthermore, we consider the maximum background knowledge available to the adversary and focus on a Bayesian inference attack model [35]. Namely, the adversary knows users' historical localization request probabilities of all training samples, the historical training samples, and their current localization requests.

4.3 Adp-FSELM algorithm

As shown in Fig. 3, the Adp-FSELM algorithm consists of four operation phases: labeled sample obfuscation, samples feature aggregation, differentially private feature fusion, and differentially private model training. In this section, we will discuss how noise is added to each operation phase so that location privacy is preserved. Since the Adp-FSELM consists of three private operation phases, the total privacy budget ε is divided into three sub-budgets ε_1 to ε_3 , such that $\sum_{i=1}^3 \varepsilon_i = \varepsilon$. Detailed discussion on privacy analysis is presented in the next section.

4.3.1 Labeled samples obfuscation

In this phase, assume that each user with the edge devices holds a RSSI fingerprint dataset X , which is collected in an interested area from a building B . By using feature extraction for X , the user obtains a RSSI signal feature vector X_N , the X_N contains l labeled sample features $X_L = \{(x_i, t_i) | x_i \in \mathbf{R}^{n_1+n_2}, t_i \in \mathbf{R}^m, i = 1, 2, \dots, l\}$ and u unlabeled sample features $X_U = \{(x_i) | x_i \in \mathbf{R}^{n_1+n_2}, i =$

$1, 2, \dots, u\}$ from different APs in the building B , where $x_i = [x'_i, x''_i]$ is composed of n_1 BLE RSSI feature measurements and n_2 WiFi RSSI feature measurements. To enjoy the desired indoor location-based services, edge user randomly sends his $X_N = [X_L, X_U]$ to edge node to aggregate the features. Considering that any edge node is semi-trusted and may have the risk of privacy disclosure when processing RSSI training samples, we inject noise into the labeled RSSI features in advance before they are sent to edge nodes. In this way, the performance of privacy protection can be further strengthened [28].

Specifically, each user breaks his RSSI training samples X_N into two parts for WiFi and BLE RSSI data, respectively: labeled samples X_L and unlabeled samples X_U . To obfuscate the correspondence between the X_L and its labels T , we add the appropriate noise $N_0/N \times \text{Laplace}(\Delta f/\varepsilon_1)$ to the X_L to generate the obfuscated RSSI samples X'_L , which can be represented as:

$$X'_L = X_L + N_0/N \times \text{Laplace}(\Delta f/\varepsilon_1), \quad (15)$$

where N_0 indicates the amount of labeled RSSI training samples, N_0/N is the label ratio of the RSSI training samples, and ε_1 is the sub-budget. Intuitively, the labeled samples may disclose more useful information to the edge nodes. Therefore, we add appropriate noise to the labeled samples to hide the participation of a single individual. According to Definition 3, the X'_L satisfies ε_1 -differential privacy. It is noted that the global sensitivity Δf gives an upper bound on how much we must perturb its output to preserve privacy and the label ratio N_0/N dynamically fine-tunes the amount of noise addition. The Δf is determined by the Manhattan distance between maximum value and minimum value of the average received signal strength of n_1 BLE RSSI measurements and n_2 WiFi RSSI measurements at m locations, that is, the $\Delta f = \|\max_m(x_{i,\text{avg}}) - \min_m(x_{i,\text{avg}})\|_1$, where $x_{i,\text{avg}}$ is the mean of n_1 BLE RSSI measurements and n_2 WiFi RSSI measurements at each location t_i .

4.3.2 Samples feature aggregation

After receiving the obfuscated RSSI samples X'_L and unlabeled RSSI samples X_U of WiFi and BLE from edge devices, the edge nodes first aggregate these RSSI samples by selecting the same position in which the WiFi or BLE RSSI vector is obtained in the same area, and then send them to the edge servers.

In this situation, edge nodes merely undertake the simple and lightweight feature aggregation. In addition, no edge nodes can acquire or integrate a complete training RSSI samples because they cannot be colluded with each

other. Moreover, edge nodes can hardly compromise the privacy of X'_L due to the nature of differential privacy [32].

Specifically, by combining X'_L and X_U , the edge nodes can obtain the differentially private training samples dataset $[T [X'_L X_U]]$.

4.3.3 Differentially private feature fusion

After receiving the aggregated training sample dataset $[T [X'_L X_U]]$, in order to prevent the privacy leaks of fusion information of WiFi and BLE fingerprints for graph Laplacians, the edge servers calculate the differentially private graph Laplacians L_1 and L_2 , respectively, by adding appropriate noise to BLE and WiFi fingerprints. The fusion operation can also be replaced with the element-wise, summation, multiplication, or other advanced fusion techniques.

In order to comprehensively consider the minimization of structural risk, the complexity of the model, and privacy protection of fusion information, we add suitable noise determined by $\text{Laplace}(2\Delta f/\varepsilon_2)$ to the graph Laplacians L_1 and L_2 in Formula (9), respectively. Thus, the target function of the Adp-FSELM can be represented as

4.3.4 Differentially private model training

After obtaining the β , the cloud server builds the RSSI fingerprints map for the localization model $f = H\beta$. Since cloud server is untrusted, it may infer the terminal users' location privacy by analyzing the training parameters (e.g., a_k and b_k) [2, 34].

For generating the differentially private output matrix, we inject noise generated by $\text{Laplace}(\Delta f/\varepsilon_3)$ into activation function $g(x)$ and thus obtain

$$G'(a_k, b_k, x_i) = g(a_k \cdot x_i + b_k + \text{Laplace}(\Delta f/\varepsilon_3)), \quad (17)$$

where $k = 1, 2, \dots, L$. In the Adp-FSELM, we use $G'(a_k, b_k, x_i)$ as the output of the k_{th} hidden node in regard to the input RSSI x_i . This operation phase aims to obfuscate the final published locations to prevent cloud server from mining users' location privacy.

In the Adp-FSELM, each private operation phase provides ε_i -DP according to Definition 3. Thus, the output of our network strictly satisfies ε -differential privacy as a whole, which guarantees that the Adp-FSELM is differentially private.

The formal description of the Adp-FSELM can be summarized as Algorithm 1.

Algorithm 1 DP-FSELM

Input: The original training samples $[T [X_L X_U]]$; the activation function $G'(x)$; graph Laplacians L_1 and L_2 ; coefficients λ_1 and λ_2 ; the number of hidden nodes L initial $\varepsilon_1, \varepsilon_2, \varepsilon_3$.

Output: The localization prediction f .

- 1: Randomly generate the values for weight a_k and bias b_k , $k = 1, 2, \dots, L$;
 - 2: Add noise $N_0/N \times \text{Laplace}(\Delta f/\varepsilon_1)$ to X_{L2} and aggregate the private training dataset $[T [X'_L X_U]]$;
 - 3: Separately calculate the graph Laplacians L_1 and L_2 and add noise $\text{Laplace}(2\Delta f/\varepsilon_2)$, to them respectively;
 - 4: Generate perturbed output matrixes H^* through $G'(x)$ obtained by adding noise $\text{Laplace}(2\Delta f/\varepsilon_3)$;
 - 5: Calculate $\beta = ((J + \lambda_1 L_1^T + \lambda_2 L_2^T) H^*)^\dagger J T$;
 - 6: Use $f = H\beta$ to generate the localization prediction.
-

$$\arg_{\beta} \min \left\{ \frac{1}{2} \|JH\beta - T\|^2 + \lambda_1 (H\beta)^T L_1 H\beta + \text{Laplace}\left(\frac{2\Delta f}{\varepsilon_2}\right) + \lambda_2 (H\beta)^T L_2 H\beta + \text{Laplace}\left(\frac{2\Delta f}{\varepsilon_2}\right) \right\}. \quad (16)$$

Obviously, by solving the optimization problem in Formula (16), we obtain the solution for β while preventing the privacy leaks during the fusion of the WiFi and BLE RSSI fingerprints.

4.3.5 Privacy analysis

We first introduce sequential composition properties of DP protection [36]:

Lemma 1 (Sequential composition) *Given random algorithms $\mathcal{M}_1, \mathcal{M}_2, \dots, \mathcal{M}_n$ on the same dataset D and its corresponding privacy budget ε_1 , the combination of these algorithms provides $\sum_{i=1}^n \varepsilon_i$ -DP protection for whole training process.*

Theorem 1 *The Adp-FSELM is ε -differentially private.*

Proof According to Algorithm 1, the Adp-FSELM contains three private operation phases: (1) labeled samples

obfuscation (cf. Step 2 in Algorithm 1), (2) private fingerprint fusion (cf. Step 3 in Algorithm 1), and (3) private output generation (cf. Step 4 in Algorithm 1). According to Definition 3, each individual private operation phase satisfies ε_i -DP. Applying Lemma 1, we can obtain that the Adp-FSELM preserves $\sum_{i=1}^n \varepsilon_i$ -DP. Recall that ε is total privacy budget for the Adp-FSELM and is distributed to three sub-privacy budgets such that $\sum_{i=1}^3 \varepsilon_i = \varepsilon$. Since each private operation phase in the Adp-FSELM consumes the sub-privacy budget ε_i allocated to it, according to Lemma 1, the total privacy budget ε will be depleted after the differentially private model training phase is complete. Note that Lemma 1 holds even when subsequent computation incorporates the outcomes of preceding computations, because the Adp-FSELM uses the DP-based activation function and graph Laplacians parameters. That is, for any two adjacent training datasets X and X' , $G'(x)$ satisfies

$$P[G'(X) \in \mathcal{S}] \leq e^\varepsilon \cdot P[G'(X') \in \mathcal{S}] \quad (18)$$

Moreover, according to Sect. 4.2, no edge nodes can acquire or integrate complete training RSSI samples because they only use part of unlabeled and perturbed labeled RSSI samples. Furthermore, the Adp-FSELM prepares several differentially private parameters and performs noise addition at each private operation phase. Therefore, when the edge nodes aggregate the perturbed samples without accessing the raw RSSI samples, the following process will provide ε -DP. Consequently, Adp-FSELM is ε -differentially private.

We can observe that the location accuracy of the Adp-FSELM is related to privacy budget ε . In order to guarantee the location privacy of the whole location estimation model f , we added different Laplace noise (determined by ε_1 , ε_2 , ε_3) into three private operation phases, respectively. Allocation of ε to ε_i can be implemented either by Adp-FSELM automatically or according to the terminal users' specification. Actually, according to Lemma 1, the Adp-FSELM can preserve ε -DP, regardless of how the sub-privacy budget ε_i is distributed to these private operation phases. However, the allocation strategy may affect the location accuracy. In the current implementation of Adp-FSELM, we employ a privacy budget allocation rule that we empirically determine to obtain high average location accuracy: $\varepsilon_1 = \varepsilon/4$ for the label sample obfuscation phase, $\varepsilon_2 = \varepsilon/2$ for the private fingerprint fusion phase, and $\varepsilon_3 = \varepsilon/4$ for the private output generation phase. From this, one can see that not all private operation phases need high privacy budgets to ensure accuracy. Allocating unnecessary high privacy budgets to such private operation phases will negatively affect location accuracy because it would steal from those private operation phases that do need high

privacy budgets to remain accurate. One way for further optimizing the allocation strategy of privacy budget is to use regression-based learning on ε [37]. This is a plan of our future work.

Although current privacy budget allocation strategy of Adp-FSELM is beneficial from the perspective of maximizing location accuracy, we still need to discuss its privacy implications. Since each private operation phase of Adp-FSELM ε_i -DP, a larger ε_i will make the RSSI signal feature be accurately retained, whereas a lower ε_i will induce it to be more perturbed. A variable privacy budget allocation strategy allows the user to distribute ε in a way that reflects which private operation phase he/she perceives is more sensitive and should be more obfuscated to protect its privacy. For example, if users feel that label sample obfuscation phase is more sensitive, then a lower ε_1 can be distributed to this private operation phase. If private output generation phase is more sensitive, then a lower ε_3 can be allocated to this private operation phase, thus resulting in a more perturbation on active function. This provides flexible privacy requirements in terms of what needs to be preserved for different perceptions.

Lemma 2 *A scheme can resist the inference attack if attackers cannot successfully identify the user's real location from user's localization request information.*

Theorem 2 *The Adp-FSELM is Bayesian inference attack [35] resistant under ε -DP.*

Proof According to the privacy threat model (see Sect. 4.2), we assume an adversary knows the probabilistic distribution of a user in actual region t , denoted by $\pi(t)$, the model result probability $P[t, t^*]$ for any actual region t and target region t^* . When the adversary observes the t^* , he will use Bayes' rule to predict a probability distribution of the user's location, which can be expressed as follows:

$$\sigma(t) = \frac{P[t, t^*] \cdot \pi(t)}{\sum_{t' \in \mathcal{R}} P[t', t^*] \cdot \pi(t')}. \quad (19)$$

The Adp-FSELM can effectively limit the knowledge gain of the attacker, i.e., $\sigma(t)/\pi(t)$ in the previous assumption based on the principle of ε -differential privacy. Based on Theorem 1, whatever the adversary's prior knowledge $\pi(t)$ is, his posterior knowledge σ satisfies:

$$1/e^\varepsilon \leq \sigma(t)/\pi(t) \leq e^\varepsilon. \quad (20)$$

Intuitively, the adversary cannot figure out whether the actual region is t or t' . Closer probability and lower value of ε significantly contribute to more powerful privacy guarantee. In this way, the ADP-FSELM algorithm can guarantee that there are enough positioning requests whose historical localization request probabilities are as same as that of the user's real localization model. Thus, the

adversary is unable to obtain the user's real location and localization samples by reversing the positioning algorithm.

Therefore, the Adp-FSELM is Bayesian inference attack resistant under ε -DP.

5 Experimental evaluation

5.1 Experimental setup

5.1.1 Conditions and datasets

In this section, we conducted comprehensive simulation experiments to evaluate the performance of the Adp-FSELM. In order to simulate the edge-cloud collaboration system, we rent a Baidu cloud as the cloud server of Adp-FSELM and realize the edge server and edge device by simulation on our server and personal computer. A WiFi router and socket protocol are used to simulate the communication process between entities of the Adp-FSELM, and all algorithms are implemented by using MATLAB software. The experimental environments and setup are shown in Table 1.

We evaluated the performance of Adp-FSELM by using two real datasets presented in the literature [8]. The datasets were collected by smartphones, which were well equipped with WiFi and BLE APs. The floorplan of the dataset is shown in Fig. 4, where the Office Area dataset has dimensions of $12.5 \times 7.5 \text{ m}^2$ and the Mall Area dataset has dimensions of $31.8 \times 14.95 \text{ m}^2$. Any missing RSSI measurements were set to a minimum signal strength value -95 dBm in the test environment. For more detailed information on datasets collection, interested readers can refer to the literature [8].

Office Area dataset A total of 3916 samples were collected from 104 data acquisition points in the Office Area. Around all data acquisition points, there were 20 BLE beacons and 30 WiFi APs detected. Thus, each sample includes location (x, y) and 50-dimensional signal features (20-dimensional BLE and 30-dimensional WiFi).

Mall Area dataset A total of 9492 valid samples were collected from 480 data collection points in the Mall Area. Around all collection points, there were 22 BLE and 35 WiFi APs presented. Thus, each sample includes location (x, y) and 57-dimensional signal features (22-dimensional BLE and 35-dimensional WiFi).

The BLE and WiFi signal features were spliced together in each dataset. In addition, all RSSI samples are normalized with z-score before being used.

5.1.2 The location accuracy metric

According to [8], the location accuracy varies under different error distance conditions in ML-based indoor positioning estimation. For example, given a well-trained location estimation model f and a testing RSSI sample $\{X_t, Y_t\}$, where X_t is the signal feature vector and Y_t is the real location, an estimated location $Y' = f(X_t)$ can be obtained. The estimated error distance between Y_t and Y' can be computed though the Euclidean distance $ed = \|Y' - Y_t\|_2$, which could be used to evaluate the accuracy of localization model. Obviously, it is not efficient because how much ed indicates the high accuracy of location model is not known. Therefore, in order to effectively evaluate the performance of the Adp-FSELM, we will discuss the location accuracy in regard to a certain tolerance error distance threshold ED. If ed is less than ED, then the estimated location is considered to be a correct position, otherwise it is taken as an incorrect position. As illustrated in Fig. 5, the estimated error distance between Y'_1 and the real location is less than ED_1 and the estimated distance between Y'_2 and the real location is less than ED_2 , so they are considered to be the correct results with respect to the ED_1 and ED_2 , respectively. Since the estimated distance for Y''_1 and Y''_2 is larger than ED_1 and ED_2 , respectively, they are considered to be wrong results. Therefore, we define the location accuracy regarding a certain error distance threshold ED as the ratio of the number of estimated correct location to the number of input locations.

Table 1 The experimental environments and setup

Entities	Hardware specifications	Software specifications
Cloud server	(1) CPU: Intel(R) Xeon(R) Gold 6148@2.40 GHz (2) RAM: 4GB	(1) OS: Windows Server 2019 Datacenter (2) Software: MATLAB 2018b
Edge server	(1) CPU: Intel(R) Core i7-8700k@3.2 GHz (2) RAM: 16GB	(1) OS: Ubuntu 18.04.2 (2) Software: MATLAB 2018b
Edge device	(1) CPU: Intel(R) Core i7-8750H@2.4 GHz (2) RAM: 24 GB	(1) OS: Windows 10 Enterprise Edition (2) Software: MATLAB 2018b
WiFi	TP-LINK TL-WAR450L	

Fig. 4 Floor plan of the experimental site **Fig. 5** Evaluation of location accuracy based on error distance **Fig. 6** Time cost of the cloud-only system

5.2 Examining of edge–cloud collaboration system in fingerprint localization

In this section, we investigate the capacity of the cloud-only system and edge–cloud collaboration system in fingerprint indoor localization regarding latency and location accuracy.

5.2.1 Time cost of the cloud-only paradigm in fingerprint localization

Figure 6 shows the time costs of the Adp-FSELM in the phases of data transmission, data fusion, and model training on the Office Area and Mall Area datasets. Since the time overhead on fingerprints feature aggregation phase is very small and almost negligible, we present the time overheads in RSSI data transmission phase, RSSI features fusion phase, and localization mode training phase. It can be observed that no matter on which dataset, the time cost in RSSI features fusion phase is the most, while the time spent in RSSI data transmission phase is the least, and the time overhead in localization model training phase is between two of them. Moreover, as the amount of RSSI data and data features increase, the time cost required for each operation phase is greatly increased. For example, on Office Area dataset with 3916 samples and 50-dimensional signal features, the time cost in RSSI data transmission phase, data fusion phase, and model training phase are 2.263 s, 12.225 s, and 9.700 s, respectively; on Mall Area dataset with 9492 samples and 57-dimensional signal features, the time overheads in the corresponding operation phases are 6.162 s, 44.318 s, and 29.061 s separately. Obviously, in indoor localization wireless big data scenarios, to reduce the load on the cloud server, it is necessary for us to offload the feature fusion phase to edge services. This also demonstrates that our Adp-FSELM algorithm is reasonable.

5.2.2 The latency and accuracy of edge–cloud collaboration paradigm in fingerprint localization

For the purpose of illustrating the performance of edge–cloud collaboration fingerprint IPS in terms of latency (time overhead), as shown in Fig. 7, we compare the localization systems' testing data transmission latency, localization model testing latency, and localization results return latency in terms of different numbers of RSSI samples on Office Area dataset and Mall Area dataset in

Fig. 7 Latency of indoor localization testing on different datasets

Fig. 8 Evaluation of location accuracy based on error distance

cloud-only system and edge–cloud collaboration system. Similarly, we also compare the location accuracy of indoor localization on different datasets with different dimensional signal feature in Fig. 8.

As can be seen from Fig. 7a, the total latency of online positioning process for the Adp-FSELM in cloud-only system on Office Area dataset is much greater than that in edge–cloud collaboration system. For example, for one testing sample, the total latency in cloud-only system is 353 ms, while the total latency in edge–cloud collaboration system is 70 ms. The total latency in cloud-only system is more than 5 times that in edge–cloud collaboration system. Moreover, as the number of test samples increases, the overall latency of online positioning for the Adp-FSELM increases gently. Another key observation is that the data transfer latency (containing test set transfer latency and test results transfer latency) is higher than localization model computation latency.

Similar to the trends shown in Fig. 7a, Fig. 7b shows that the data transfer latency is always higher than model computation latency, and edge–cloud collaboration system in fingerprint localization is superior to cloud-only system in fingerprints indoor positioning in terms of latency, especially on large samples set.

Figure 8 shows the location accuracy of the Adp-FSELM under $ED = 10m$ on different datasets: Office_WiFi, Office Area, Mall_WiFi, and Mall Area. The Office_WiFi dataset contains 3916 samples with only 30-dimensional WiFi signal features, which is derived from the corresponding Office Area dataset. Similarly, the Mall_WiFi dataset is derived from the Mall Area dataset and contains 9492 samples with only 30-dimensional WiFi signal features. As can be seen from Fig. 8, the richer the data features (e.g., the Officer_WiFi against the Office Area, and Mall_WiFi against Mall Area), the higher the training accuracy and testing accuracy of the Adp-FSELM. However, when the data volume increases (e.g., the Officer_WiFi against the Mall_WiFi, and Office Area against Mall Area), the training accuracy and testing accuracy of localization model hardly change. This shows that the richness of signal features has greater impact on the location accuracy of model than the amount of data samples. In the edge–cloud collaboration system, the features and types of collected RSSI data will be richer due to the existence of a large amount of edge devices and edge nodes. However, in cloud-only system, it is difficult to ensure the diversity of data features and types. To this end, compared to the cloud-only system, the edge–cloud collaboration in fingerprint localization system has advantages in terms of location accuracy.

In conclusion, compared to the cloud-only system for fingerprint localization, the edge–cloud collaboration system for fingerprint localization has great advantages

regarding latency and location accuracy, which is more suitable for fingerprint indoor localization.

5.3 Performance evaluation

In the literature [8], we achieve an optimal partition of the training and testing samples by extensive experiments. To compare fairly with the state-of-the-art methods under the same conditions, we adopt the same experimental setup and data partition strategy as FSELM used. The Office Area dataset is randomly divided into training samples (3000 data) and testing samples (916 data) for training model. In the Mall Area dataset, a total of 4800 samples are used as the training dataset and the remaining 4692 samples as the testing dataset.

We evaluated performance of the Adp-FSELM from four aspects: the impact of privacy budget, semi-supervised trusted learning performance, location accuracy comparison, and time consumption. It is noted that the number of network hidden nodes is 1000, the optimal pair of constraint coefficients (λ_1 , λ_2) is set to (0.2, 0.05), and the nonlinear sigmoid function is selected as the activation function. The optimal parameters of other semi-supervised manifold learning methods are determined by using the same method as those in the literature [8]. In addition, the cross-validation is applied to determine the manifold constraint coefficients, and each result is an average value of 100 training sessions.

5.3.1 Performance evaluation on the office area dataset

• The impact of privacy budget

Since the Adp-FSELM provides ϵ -DP guarantee, it inevitably results in a loss of location accuracy. That is, the setting of parameter ϵ affects the location accuracy of the Adp-FSELM, as the scale of Laplace noise is determined by ϵ . To evaluate the impact of total privacy budget ϵ on the performance of the Adp-FSELM, we used all the labeled training samples to calculate the location accuracy

Fig. 9 Location accuracy of Adp-FSELM with different ϵ

Fig. 10 Error distance of the DP-FSELM with different privacy budgets

regarding a certain ED, and error distance ed by varying ϵ from 0.00001 to 1. The results are shown in Figs. 9 and 10.

Figure 9 shows the location accuracy with different total privacy budgets ϵ . Obviously, location accuracy increases with the increment of ϵ . This is because that the bigger the ϵ , the less noise is added in this operation phase, thus leading to high location accuracy. It is interesting that, when $\epsilon \geq 0.1$, the location accuracy regarding a certain ED is almost constant. This shows that, by adding appropriate noise, the Adp-FSELM can achieve good location accuracy while preserving user's location privacy. Figure 9 also shows that under the same privacy budget ϵ , larger ED corresponds to higher location accuracy. For example, with $\epsilon = 0.1$, the location accuracy of the Adp-FSELM is nearly 100% when $ED = 5m$, while it gets 96.2% when $ED = 3m$. The reason is that with the same noise added in each private operation phase, the larger ED means that there are more perturbed locations outputted by localization estimation model f that satisfy the condition $ed \leq ED$ and are more likely to be correct results. However, with the parameter $\epsilon < 0.1$, the location privacy is still poor even if $ED \geq 3m$. Therefore, it is necessary for the Adp-FSELM to trade off the privacy and location accuracy to meet the users' privacy requirements and service experience. Figure 10 shows the error distance ed with different total privacy budget ϵ (e.g., $\epsilon \in [0.001, 1]$).

As can be seen from Fig. 10, the error distance ed decreases with the increment of the privacy budget ϵ , and the largest value of error distance ed is equal to 3.582 m with $\epsilon = 0.001$. In Fig. 9, with parameter $\epsilon = 0.01$, the accuracy of localization model reaches close to 90% regarding $ED = 4m$, which shows the correctness of our algorithm. In addition, error distance ed declines slightly and is almost stable when $\epsilon \geq 0.1$. With the parameter $\epsilon = 0.1$, error distance ed gets 1.224 m, that is high location

accuracy. According to Definition 1, smaller ε guarantees higher privacy, which can also lead to poor location accuracy and more time consumption. Therefore, in order to trade off location privacy, location accuracy, and time consumption, we set ε as 0.1 for the following experiments.

• Semi-supervised trusted learning performance

As there is no prior work on privacy-preserving for fingerprint-fusion-based indoor localization in edge computing, we simply compare the Adp-FSELM algorithm with the FSELM and several semi-supervised learning methods (e.g., the LeMan [22], the SELM [16], the S-RVFL, and the FS-RVFL [8]) with $\varepsilon = 0.1$, on the 3000 pre-selected training samples. The number of labeled samples N_0 varies from 100 to 3000. Based on manifold regularization, LeMan solves the problem of tracking the trajectory in a predefined area equipped with a sensor network when most of the coordinates on the trajectory are unknown. Both the S-RVFL and FS-RVFL [8] are constructed based on the RVFL [23]. The RVFL relies on the learning and generalization characteristics of random vector functional link net, which considers not only links from the enhanced pattern to the outputs but also directly links from the inputs to the outputs.

In the following experiments, all the location accuracies will be discussed regarding a certain ED (from 1m to 5m) and the parameters of the methods mentioned above are shown in Table 2.

The results are shown in Figs. 11 and 12. As N_0 increases, the location accuracy of all six methods also increases. For example, in Fig. 11, the accuracy of Adp-FSELM at $N_0 = 1000$ is almost 30% higher than that at $N_0 = 100$ regarding ED = 2m. As shown in Fig. 11, when the N_0 is less than 50% of the total number of samples, positive labeled samples are insufficient. Hence, we added less noise to guarantee the performance of the model training. The experimental results demonstrate that compared with the other four methods, the performance of the Adp-FSELM and the FSELM is always similar to each other, and both of them almost achieve the highest accuracy, that is, the Adp-FSELM controls the loss of the location accuracy while guaranteeing privacy. This is

probably because we locally added the appropriate Laplace noise at each operation phase of the training process.

With the increasing N_0 , increased labeled samples were input into the training network for contributing to the model training, and the FSELM, SELM, LeMan, S-RVFL, and FS-RVFL obtain enough training data to learn the RSSI features and reach the stable state with high accuracy. Accordingly, the noise $N_0/N \times \text{Laplace}(1/\varepsilon_1)$ added to training samples also increases gradually. Therefore, as shown in Fig. 12, when N_0 is close to the total number of samples, the Adp-FSELM cannot maintain the outstanding performance against these comparative methods, but it still provides acceptable location accuracy. For example, when $N_0 = 1500$, the top gap between the location accuracy of FSELM and Adp-FSELM is almost 18%. It is because that the increment of the labeled training data incurs the higher demand for privacy protection, and more noise should be added to this operation phase to ensure the credible training process. So, the deterioration of Adp-FSELM's location accuracy became visible.

From $N_0 = 2000$ to 3000, the location accuracy of Adp-FSELM starts rising. It might be the contribution of labeled samples that exceed the influence of noise for the model training. Fortunately, the mean absolute error (MAE) of total location accuracy between the Adp-FSELM and the FSELM can still be restricted to 2.22%.

• Location accuracy comparison

We compare the testing accuracy of the Adp-FSELM with the FSELM for different proportions of RSSI calibration data.

We evaluate the location accuracy regarding different ED (e.g., $\text{ED} \in \{1m, 2m, 3m, 4m, 5m\}$). The training data were also given as mentioned above for model training, and then the model was tested on 916 test data. Figure 13 shows the prediction accuracy of the localization model for different numbers of N_0 (e.g., from 100 to 3000).

As can be seen from Fig. 13, we can conclude that as the amount of labeled calibration data increases, the Adp-FSELM and the FSELM models can achieve an increasingly high level of testing accuracy. This trend is continued even if the histograms are extended beyond the plotted

Table 2 Parameters of the semi-supervised manifold learning methods for Office Area dataset

Method	λ	#nodes	Active function
LeMan [22]	$\lambda = 1$	1000	—
FSELM [8]	$\lambda_{\text{BLE}} = 0.2, \lambda_{\text{WiFi}} = 0.05$	1000	'sig'
Adp-FSELM [ours]	$\lambda_{\text{BLE}} = 0.2, \lambda_{\text{WiFi}} = 0.05$	1000	'sig'
SELM [16]	$\lambda = 0.3$	1000	'sig'
FS-RVFL [8]	$\lambda_{\text{BLE}} = 0.3, \lambda_{\text{WiFi}} = 0.3$	1000	'sig'
S-RVFL [8]	$\lambda = 0.7$	1000	'sig'

Fig. 11 Semi-supervised learning performance of Adp-FSELm with the N_{f0} is less than 50% of the total number of samples

Fig. 12 Semi-supervised learning performance of Adp-FSELm with the N_0 is close to the total number of samples

Fig. 13 Location accuracy of the FSELm and Adp-FSELm with different N_0

ranges and reach a peak. For example, with $ED = 3m$, both of the models reach up to more than 85% after $N_0 = 500$. When N_0 is larger than 50% of the total number of training samples (e.g., $N_0 \geq 1500$), the Adp-FSELm model achieves more than 90% location accuracy. That is, there is no significant loss of utility for the Adp-FSELm though adding appropriate noise in each operation phase. Therefore, the Adp-FSELm makes FSELm's training process and generated localization model provably security.

It is interesting that in some cases (e.g., when $N_0 = 100$ and $ED = 3m/4m$), the localization performance of the Adp-FSELm exceeds the FSELm. This is probably because the training process of these two methods is random and dynamic, which gradually converges through the network. The execution of privacy protection mechanism needs to add noise satisfying DP locally to each private operation phase of the Adp-FSELm, which may cause some positive or negative fluctuations in location accuracy.

• Time consumption comparison

Due to the resource limitation of edge nodes and smart edge devices, we need to take the time consumption of the localization process into account. For a constant training set of size 3000, we evaluate the time consumption, while the number of labeled samples N_0 varies from 100 to 3000.

Figure 14a shows the average training time consumption of 100 sessions of seven localization methods: Adp-FSELm, FSELm, SELm, FS-RVFL, S-RVFL, ELM, and support vector machine (SVM). It shows that the training time consumption of the ELM and SVM increases with the increment of N_0 , because these two methods only use labeled data for model training, and more labeled samples will require more time to process. By contrast, the Adp-FSELm and the other four semi-supervised approaches (e.g., FSELm, SELm, FS-RVFL, and S-RVFL) maintain an almost constant time consumption. Particularly, compared with the FSELm, the Adp-FSELm has almost no

(a) Training time consumption

(b) Testing time consumption

Fig. 14 Training and testing time consumption

significant increase in time consumption. This is mainly because that we locally added the appropriate noise to each operation phase of the model training process instead of simply adding a global noise to the whole output, which naturally has little effect on computational complexity.

Figure 14b shows the testing time on the 916 testing samples. Compared with the SELM, the FS-RVFL, and the S-RVFL, the Adp-FSELM does not have much extra computation overhead and thus performs outstanding performance on both model training and testing. The main reason for this phenomenon is that the Adp-FSELM inherits the beneficial properties of ELM regarding and testing speed. Compared to the ELM, the SVM, and the FSELM, our method achieves almost closed time consumption. It demonstrates that adding appropriate noise to each private operation phase has little effect on time consumption. Therefore, the Adp-FSELM is a lightweight privacy-preserving mechanism for resource-constrained edge computing networks.

5.3.2 Performance evaluation on the mall area dataset

We also evaluate performance of the Adp-FSELM on the Mall Area dataset that is larger than the Office Area dataset. Like the experiments for Office Area dataset, we optimize the parameters of all semi-supervised manifold learning methods for Mall Area dataset and use the mean of location accuracy under the error distance threshold ED from $1m$ to $10m$ to select the global parameters, which are shown in Table 3.

In the experiments, we allocate a different total privacy budget ϵ to verify the effectiveness of the Adp-FSELM. Firstly, we study the influence of total privacy budget ϵ on location accuracy with respect to a certain ED (e.g., $ED \in \{1m, 2m, \dots, 10m\}$). The results are shown in Fig. 15.

Like in Fig. 9, the location accuracy increases with the increment of ϵ , and under the same privacy budget ϵ , larger ED corresponds to higher location accuracy. For example, with $\epsilon = 0.01$, the location accuracy of the Adp-FSELM is 97.8% when $ED = 7m$, while it gets 87.4% when $ED = 5m$. In addition, when $\epsilon \geq 0.01$ and $ED \geq 3m$,

Fig. 15 Influence of different privacy budget on location accuracy

location accuracy is constantly approaching to 100%. Therefore, in order to trade off the privacy and localization accuracy, we set privacy budget $\epsilon = 0.01$ for Mall Area dataset.

Then, we compare performance of various semi-supervised manifold learning methods using the 4800 training samples and 4692 testing samples. The parameter N_0 varies from 100 to 4000, and the error distance threshold ED varies from $1m$ to $10m$. According to Fig. 15, when $ED \geq 3m$ or $4m$, the location accuracies are almost equal and close to 100%. Besides, when error distance threshold ED is equal to $3m$ or $4m$, we can clearly distinguish the different shops for Mall Area dataset. Moreover, it is an acceptable error distance in indoor environments. Therefore, in Tables 4 and 5, we only present the results with respect to $ED = 3m$ or $4m$.

As we can see from Table 4, the FSELM still performs outstanding location accuracy among these semi-supervised learning methods on Mall Area dataset. When $\epsilon = 0.0001$ and 0.001 , the Adp-FSELM has the worst location accuracy. However, the performance of the Adp-FSELM is very close to FSELM when ϵ is equal to 0.01 and 0.1 . From Table 5, we can observe that the MAE of the accuracy of the training and testing process significantly increases with the decrease in ϵ . The reason lies in that when ϵ decreases, the noise added at each operation phase of the training process is large, which causes a large MAE. For example,

Table 3 Parameters of the semi-supervised manifold learning methods for Office Area dataset

Method	λ	#nodes	Activation function
LeMan [22]	$\lambda = 1$	1000	—
FSELM [8]	$\lambda_{BLE} = 0.2, \lambda_{WiFi} = 0.05$	1000	'sig'
Adp-FSELM [ours]	$\lambda_{BLE} = 0.2, \lambda_{WiFi} = 0.05$	1000	'sig'
SELM [16]	$\lambda = 0.2$	1000	'sig'
FS-RVFL [8]	$\lambda_{BLE} = 0.4, \lambda_{WiFi} = 0.4$	1000	'sig'
S-RVFL [8]	$\lambda = 0.9$	1000	'sig'

Table 4 Manifold learning accuracy and location accuracy with ED = 3m or 4m

N_0	100	300	500	1000	2000	3000	4000
Training accuracy/% (ED = 3m/4m)							
LeMan	38.7/55.9	44.7/65.9	46.8/67.9	52.5/73.8	61.3/81.8	67.0/85.8	70.9/89.0
S-RVFL	40.0/58.7	51.7/71.6	59.5/79.1	69.8/86.2	80.5/93.8	86.2/95.8	86.2/95.8
SELM	54.7/73.4	67.0/83.4	73.5/86.4	84.0/93.2	89.0/96.2	91.9/96.7	93.9/97.4
FS-RVF	43.5/65.4	68.6/87.3	77.9/89.8	85.9/94.1	90.1/94.1	92.4/96.6	94.4/98.2
FSELM	69.0/87.3	79.7/91.7	83.4/93.9	87.4/94.4	90.9/96.5	93.8/97.9	95.1/98.8
Adp-FSELM							
$\varepsilon = 0.1$	66.9/85.8	79.5/91.2	81.4/92.1	84.7/93.4	90.7/96.2	93.5/97.6	95.3/98.6
$\varepsilon = 0.01$	63.5/81.2	75.9/88.6	79.3/90.6	82.3/92.9	90.3/95.9	93.7/97.9	94.7/98.4
$\varepsilon = 0.001$	46.1/65.7	56.0/77.0	72.9/83.9	75.2/83.8	78.1/84.1	83.4/90.7	91.9/95.3
$\varepsilon = 0.0001$	0.0/0.1	5.0/8.7	0.2/0.3	0.0/0.0	2.1/4.5	0.4/0.7	0.0/0.0
Testing accuracy/% (ED = 3m/4m)							
S-RVFL	40.3/55.6	51.6/72.5	57.0/75.8	68.2/85.2	78.4/90.8	82.8/93.0	85.6/94.3
SELM	52.5/72.9	62.0/80.6	68.9/83.7	78.8/89.5	84.6/92.5	86.8/94.5	88.8/95.0
FS-RVF	42.7/64.4	67.4/85.3	67.4/85.3	80.5/90.6	87.9/92.3	89.0/94.5	90.6/95.9
FSELM	69.6/87.1	77.8/90.3	79.3/91.7	81.4/92.4	84.9/92.7	88.7/95.2	90.4/95.8
Adp-FSELM							
$\varepsilon = 0.1$	64.3/83.4	74.9/87.7	77.1/88.7	79.7/89.2	84.1/91.5	88.0/94.6	88.7/95.3
$\varepsilon = 0.01$	61.9/81.7	74.6/86.2	74.4/87.4	78.6/88.9	84.3/90.4	87.3/94.2	90.0/95.1
$\varepsilon = 0.001$	44.9/62.7	44.9/61.4	57.5/68.8	67.8/77.6	40.9/58.4	53.1/73.1	68.7/79.5
$\varepsilon = 0.0001$	0.0/0.0	0.0/0.0	1.1/1.6	0.0/0.0	3.4/10.2	2.8/5.2	0.2/1.1

Table 5 The MAE and time consumption between FSELM and Adp-FSELM

Methods	ε	MAE of training accuracy/%		MAE of testing accuracy/%		Time/s	
		(ED = 3m/4m)		(ED = 3m/4m)		Training	Testing
FSELM	—	—		—		2.26	0.04
Adp-FSELM	0.10	0.70		0.95		2.27	0.15
	0.01	1.02		1.24		2.27	0.15
	0.001	4.82		7.55		2.29	0.15
	0.0001	74.82		71.85		2.86	0.15

when $\varepsilon = 0.0001$, the MAE gets more than 70%; it means that the sample's utility is broken and the localization fails. When $\varepsilon \geq 0.001$, the error of location accuracy keeps dropping as ε increases, and then it becomes gradually acceptable and constrained to within 10%. Generally, a larger ε can achieve higher location accuracy, but the MAE will be stable or rise slightly when ε reaches a threshold. When $\varepsilon = 0.01$ and $\varepsilon = 0.1$, the gap of these two conditions' MAE is very small, and the time consumption gets the minimum 2.27 s of these privacy-preserving localization comparative experiments. Therefore, the choice of $\varepsilon = 0.01$ can obtain the trade-off among data privacy, location accuracy, and time consumption.

In brief, these experiments on different fingerprint datasets indicate that the Adp-FSELM can adapt to the privacy demands of samples and can give us a hand to

control the trade-off between randomized privacy and utility, which is influenced by various parameters ε , N_0 , and others of the network. It also provides a feasible solution for privacy-preserving machine learning methods in some other applications.

6 Conclusion

In this paper, we focus on privacy-preserving problem of ML-based multi-fingerprints fusion indoor positioning in edge computing scenarios. To address this problem, we propose a differentially private fingerprint fusion semi-supervised extreme learning machine algorithm in edge computing framework, called Adp-FSELM. The Adp-FSELM employs a multi-level edge-cloud collaboration

network to meet the needs of ML-based fingerprint positioning for lightweight, low latency, real-time response, and privacy protection. As the same time, we add appropriate Laplacian noise into each operation phase of Adp-FSELM through using DP technology. Besides, we guarantee the privacy of different numbers of labeled samples for semi-supervised learning without sacrificing much location accuracy. Last but not least, we conduct the extensive experiments on practical datasets and compare our scheme with five semi-supervised learning methods on location accuracy and six mature localization methods on time consumption. The results demonstrate that the Adp-FSELM could achieve high-level privacy-preserving and acceptable location accuracy with low time consumption.

For future work, we plan to take the dynamic environment into account, where the localization model may be deteriorated with the change of APs. We need an incremental learning method to reduce the training consumption and to enhance the robustness of model. Moreover, we will study how to optimize privacy budget allocation to acquire better trade-off among privacy, location accuracy, and resource consumption.

Acknowledgements Thanks to all the reviewers and editors for their valuable comments on improving the quality of this paper, this work was partially supported by the NSFC project (Grant Nos. 61762058, 61902379, 61861024), the Natural Science Foundation of Gansu Province (Nos. 21JR7RA282, 20JR5RA404), the Foundation of A Hundred Youth Talents Training Program of Lanzhou Jiaotong University, and the Science and Technology Project of State Grid Gansu Electric Power Institute (Grant No. 52272219100P).

Declarations

Conflict of interest The authors declare that they have no conflict of interest.

Open Access This article is licensed under a Creative Commons Attribution 4.0 International License, which permits use, sharing, adaptation, distribution and reproduction in any medium or format, as long as you give appropriate credit to the original author(s) and the source, provide a link to the Creative Commons licence, and indicate if changes were made. The images or other third party material in this article are included in the article's Creative Commons licence, unless indicated otherwise in a credit line to the material. If material is not included in the article's Creative Commons licence and your intended use is not permitted by statutory regulation or exceeds the permitted use, you will need to obtain permission directly from the copyright holder. To view a copy of this licence, visit <http://creativecommons.org/licenses/by/4.0/>.

References

1. Zafari F, Gkelias A, Leung KK (2019) A survey of indoor localization systems and technologies. *IEEE Commun Surv Tutor* 21(3):2568–2599
2. Zhu XQ, Qu WY, Qiu T, Zhao LP, Atiquzzaman M, Wu DO (2020) Indoor intelligent fingerprint-based localization: principles, approaches and challenges. *IEEE Commun Surv Tutor* 22(4):2634–2657
3. Luo RC, Hsiao TJ (2018) Dynamic wireless indoor localization incorporating with an autonomous mobile robot based on an adaptive signal model fingerprinting approach. *IEEE Trans Ind Electron* 66(3):1940–1951
4. Torres-Solis J, Tiago HF, Tom C (2010) A review of indoor localization technologies: towards navigational assistance for topographical disorientation. *IntechOpen*, Rijeka
5. Li H, Sun LM, Zhu HJ, Lu X, Cheng XZ (2014) Achieving privacy preservation in WiFi fingerprint-based localization. In: *Proceedings of INFOCOM*, pp 2337–2345. <https://doi.org/10.1109/INFOCOM.2014.6848178>
6. Konstantinidis A, Chatzimilioudis G, Zeinalipour-Yazti D, Mpeis P, Pelekis N, Theodoridis Y (2015) Privacy-preserving indoor localization on smartphones. *IEEE Trans Knowl Data Eng* 27(11):3042–3055. <https://doi.org/10.1109/TKDE.2015.2441724>
7. Wang B, Chen QY, Yang LT, Chao HC (2016) Indoor smart-phone localization via fingerprint crowdsourcing: challenges and approaches. *IEEE Wirel Commun* 23(3):82–89
8. Jiang XL, Chen YQ, Liu JF, Gu Y, Hu LS (2018) FSELM: fusion semi-supervised extreme learning machine for indoor localization with Wi-Fi and Bluetooth fingerprints. *Soft Comput* 22(6):1–15. <https://doi.org/10.1007/s00500-018-3171-4>
9. NIGHTWATCHCYBER (2018) Sensitive data exposure via RSSI broadcasts in android OS. <https://www.nightwatchcybersecurity.com/2018/11/11/cve-2018-9581/>
10. Liu H, Darabi HS, Banerjee P, Liu J (2007) Survey of wireless indoor positioning techniques and systems. *IEEE Trans Syst Man Cybern Part C Appl Rev* 37(6):1067–1080. <https://doi.org/10.1109/TSMCC.2007.905750>
11. Eshun SN, Palmieri PL (2019) A privacy-preserving protocol for indoor Wi-Fi localization. *Association for Computing Machinery*, pp 380–385. <https://doi.org/10.1145/3310273.3323400>
12. Yang Z, Jarvinen K (2018) The death and rebirth of privacy-preserving WiFi fingerprint localization with paillier encryption. In: *Proceedings of INFOCOM*, pp 1223–1231. <https://doi.org/10.1109/INFOCOM.2018.8486221>
13. Wang YF, Huang MJ, Jin Q, Ma JH (2018) DP3: a differential privacy-based privacy-preserving indoor localization mechanism. *IEEE Commun Lett* 22(12):2547–2550. <https://doi.org/10.1109/LCOMM.2018.2876449>
14. Zhao P, Jiang HB, Lui JCS, Wang C, Zeng FZ, Xiao F, Li ZT (2018) P³-LOC: a privacy-preserving paradigm-driven framework for indoor localization. *IEEE/ACM Trans Netw* 26(6):2856–2869. <https://doi.org/10.1109/TNET.2018.2879967>
15. Zhu YJ, Wang Y, Liu QY, Liu Y, Zhang P (2017) WiFi fingerprint releasing for indoor localization based on differential privacy. In: *2017 IEEE 28th Annual international symposium on personal, indoor, and mobile radio communications (PIMRC)*, pp 1–6. <https://doi.org/10.1109/PIMRC.2017.8292470>
16. Liu JF, Chen YQ, Liu MJ, Zhao ZT (2011) SELM: semi-supervised ELM with application in sparse calibrated location estimation. *Neurocomputing* 74(16):2566–2572. <https://doi.org/10.1016/j.neucom.2010.12.043>
17. Gu Y, Chen YQ, Liu JF, Jiang XL (2015) Semi-supervised deep extreme learning machine for Wi-Fi based localization. *Neurocomputing* 166:282–293. <https://doi.org/10.1016/j.neucom.2015.04.011>
18. Chen JM, Wang CQ, Sun YX, Shen XM (2011) Semi-supervised Laplacian regularized least squares algorithm for localization in wireless sensor networks. *Comput Netw* 55(10):2481–2491. <https://doi.org/10.1016/j.comnet.2011.04.010>
19. Aparicio S, Perez J, Bernardos AM, Casar JR (2008) A fusion method based on Bluetooth and WLAN technologies for indoor location. In: *2008 IEEE International conference on multisensor*

- fusion and integration for intelligent systems, pp 487–491. <https://doi.org/10.1109/MFI.2008.4648042>
20. Carlos EGT, Issac G-T, Ernesto IS, Brena R (2012) WiFi blue-tooth based combined positioning algorithm. *Procedia Eng* 35:101–108. <https://doi.org/10.1016/j.proeng.2012.04.170>
 21. Carlos EG-T, Jose CC-J, Ramon FB (2013) Bluetooth-WiFi based combined positioning algorithm, implementation and experimental evaluation. *Procedia Technol* 7:37–45. <https://doi.org/10.1016/j.protcy.2013.04.005>
 22. Pan JJ, Yang Q, Chang H, Yeung D-Y (2006) A manifold regularization approach to calibration reduction for sensor-network based tracking. In: AAAI, pp 988–993
 23. Pao Y-H, Park G-H, Sobajic DJ (1994) Learning and generalization characteristics of the random vector functional-link net. *Neurocomputing* 6(2):163–180. [https://doi.org/10.1016/0925-2312\(94\)90053-1](https://doi.org/10.1016/0925-2312(94)90053-1)
 24. Shi WS, Cao J, Zhang Q, Li YHZ, Xu LY (2016) Edge computing: vision and challenges. *IEEE Internet Things J* 3(5):637–646. <https://doi.org/10.1109/JIOT.2016.2579198>
 25. Abbas N, Zhang Y, Taherkordi A, Skeie T (2017) Mobile edge computing: a survey. *IEEE Internet Things J* 5(1):450–465
 26. Satyanarayanan M, Bahl P, Caceres R, Davies N (2009) The case for VM-based cloudlets in mobile computing. *IEEE Pervasive Comput* 8(4):14–23
 27. Hu WL, Gao Y, Ha KY, Wang JJ, Amos B, Chen Z, Pillai P, Satyanarayanan M (2016) Quantifying the impact of edge computing on mobile applications. In: Proceedings of the 7th ACM SIGOPS Asia-Pacific workshop on systems, pp 1–8. <https://dl.acm.org/doi/abs/10.1145/2967360.2967369>
 28. Du M, Wang K, Xia ZQ, Zhang Y (2018) Differential privacy preserving of training model in wireless big data with edge computing. *IEEE Trans Big Data* 6(2):283–295
 29. Wang YH, Tian ZH, Su S, Sun YB, Zhu CS (2019) Preserving location privacy in mobile edge computing, pp 1–6. <https://doi.org/10.1109/ICC.2019.8761370>
 30. Huang GB, Zhu QY, Siew CK (2004) Extreme learning machine: a new learning scheme of feedforward neural networks. In: 2004 IEEE International joint conference on neural networks (IEEE Cat. No. 04CH37541), vol 2, pp 985–990. <https://doi.org/10.1109/IJCNN.2004.1380068>
 31. Wang XF, Han YW, Leung VCM, Niyato D, Yan XQ, Chen X (2020) Convergence of edge computing and deep learning: a comprehensive survey. *IEEE Commun Surv Tutor* 22(2):869–904
 32. He ZC, Zhang TW, Lee RB (2020) Attacking and protecting data privacy in edge-cloud collaborative inference systems. *IEEE Internet Things J* 8(12):9706–9716
 33. Zhang YM, Lan XL, Ren J, Cai L (2020) Efficient computing resource sharing for mobile edge-cloud computing networks. *IEEE/ACM Trans Netw* 28(3):1227–1240
 34. Zhang XJ, Chen Q, Peng XH, Jiang XL (2019) Differential privacy-based indoor localization privacy protection in edge computing. In: Proceedings of UIC, Leicester, UK, pp 491–496. <https://doi.org/10.1109/SmartWorld-UIC-ATC-SCALCOM-IOP-SCI.2019.00125>
 35. Andres ME, Bordenabe NE, Chatzikokolakis K, Palamidessi C (2013) Geo-indistinguishability: differential privacy for location-based systems. In: CCS '13. Association for Computing Machinery, New York, pp 901–914. <https://doi.org/10.1145/2508859.2516735>
 36. Dwork C, McSherry F, Nissim K, Smith A (2006) Calibrating noise to sensitivity in private data analysis. In: Theory of cryptography conference. Springer, Berlin, pp 265–284. https://doi.org/10.1007/978-3-540-32732-5_32
 37. Gursoy ME, Liu L, Truex S, Yu L, Wei WQ (2018) Utility-aware synthesis of differentially private and attack-resilient location traces. In: Proceedings of the 2018 ACM SIGSAC conference on computer and communications security, pp 196–211. <https://dl.acm.org/doi/abs/10.1145/3243734.3243741>

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Xuejun Zhang¹ · Fucun He¹ · Qian Chen^{1,2} · Xinlong Jiang² · Junda Bao¹ · Tongwei Ren³ · Xiaogang Du¹

✉ Xuejun Zhang
xuejunzhang@mail.lzjtu.cn

Fucun He
0219644@stu.lzjtu.edu.cn

Qian Chen
chenqian20b@ict.ac.cn

Xinlong Jiang
jiangxinlong@ict.ac.cn

Junda Bao
0619672@stu.lzjtu.edu.cn

Tongwei Ren
tren@wpi.edu

Xiaogang Du
duxiaogang@mail.lzjtu.cn

¹ School of Electronic and Information Engineering, Lanzhou Jiaotong University, Lanzhou 730070, China

² Institute of Computing Technology, Chinese Academy of Sciences, Beijing 100190, China

³ Computer Science Department, Worcester Polytechnic Institute, Worcester 01609, USA