SALC: Skeleton-Assisted Learning-Based Clustering for Time-Varying Indoor Localization

An-Hung Hsiao, Li-Hsiang Shen[†], Chen-Yi Chang, Chun-Jie Chiu, and Kai-Ten Feng

Department of Electrical and Computer Engineering

National Yang Ming Chiao Tung University, Hsinchu, Taiwan

[†]California PATH, Institute of Transportation Studies, University of California at Berkeley, Berkeley, USA

Email: eric28200732.cm07g@nctu.edu.tw, shawngp3@berkeley.edu, zx57012tw@gmail.com,

jack0502801.cm02g@mail.nctu.edu.tw, ktfeng@nycu.edu.tw

Abstract-Wireless indoor localization has attracted significant amount of attention in recent years. Using received signal strength (RSS) obtained from WiFi access points (APs) for establishing fingerprinting database is a widely utilized method in indoor localization. However, the time-variant problem for indoor positioning systems is not well-investigated in existing literature. Compared to conventional static fingerprinting, the dynamicallyreconstructed database can adapt to a highly-changing environment, which achieves sustainability of localization accuracy. To deal with the time-varying issue, we propose a skeleton-assisted learning-based clustering localization (SALC) system, including RSS-oriented map-assisted clustering (ROMAC), cluster-based online database establishment (CODE), and cluster-scaled location estimation (CsLE). The SALC scheme jointly considers similarities from the skeleton-based shortest path (SSP) and the time-varying RSS measurements across the reference points (RPs). ROMAC clusters RPs into different feature sets and therefore selects suitable monitor points (MPs) for enhancing location estimation. Moreover, the CODE algorithm aims for establishing adaptive fingerprint database to alleviate the timevarying problem. Finally, CsLE is adopted to acquire the target position by leveraging the benefits of clustering information and estimated signal variations in order to rescale the weights from weighted k-nearest neighbors (WkNN) method. Both simulation and experimental results demonstrate that the proposed SALC system can effectively reconstruct the fingerprint database with an enhanced location estimation accuracy, which outperforms the other existing schemes in the open literature.

Index Terms—Wireless indoor localization, clustering, timevarying, machine learning, neural networks.

I. INTRODUCTION

For decades, the emerging location-based services (LBSs) have been promoted by telecom operators which significantly relies on acquiring the position of user equipment (UE) or target devices [1]. There exist abundant techniques to be adopted for LBS, such as global positioning system (GPS) [2], passive infrared (PIR) sensors [3], WiFi [4], [5]. LBS can be adopted in a variety of contexts including indoor/outdoor localization [6] and human presence detection [7]. Nowadays, as the life-oriented demands in public areas soar, indoor LBS capable of locating a particular person or monitoring the people flow has received considerable attention. However, GPS is not suitable for indoor LBS since their signals suffer from severe environmental degradation, including scattering and blockage, which leads to unpredictably low positioning

accuracy. Therefore, short-range signal source such as WiFi becomes a potential candidate to be utilized in a complex indoor environment.

WiFi-based localization system is widely applied in indoor positioning based on WiFi access points (APs) and portable devices using received signal strength (RSS) as information inference. The RSS is the signal information related to pathloss distance between the transmitter and receiver, which can be readily obtained from WiFi APs. Fingerprinting [8]-[12] is a widely-adopted positioning algorithm based on information of APs, which typically contains both offline measurement and online estimation phases. In the offline phase, the information is measured and collected at pre-defined locations so-called reference points (RPs) to establish database consisting of RSS from APs and geometric locations of RPs. During the online phase, real-time RSS will be received and matched to those from RPs in the offline-established database to estimate the target position with the aid of signal similarity features. In [11], the authors utilize complex channel information to overcome the problems, such as data loss, noise and interference in the fingerprint database and laborious offline training. The authors in [12] have proposed a solution for alleviating frequent data collection and improve privacy in two specifically defined scenarios. Hence, fingerprinting possesses lowered computational complexity and is capable of reflecting the multipath effects of non-line-of-sight in indoor environments [13]. Moreover, the weighted k-nearest neighbor (WkNN) algorithm [14] is employed to locate the desirable device during the online phase, which is calculated based on the k largest weights among all RPs. Note that the weight implies the difference between real-time user's RSS and offline measured one in database. Consequently, it becomes important to investigate the factors disturbing RSS values which can result in acquiring faraway incorrect target location with similar RSS.

However, the RSS will also be affected by time-variation and human blockages, the authors in [15] consider the relationship between RSS of monitor points (MPs) and of RPs collected in offline phase to construct an artificial neural network for position estimation. During the online phase, the real-time RSS values at RPs are predicted based on the collected data at MPs. In general, the intention of adopting MPs is to observe the environmental changes in specific areas by continuously collecting signal information. The MPs detect the variation of RSS so that it can provide immediate information to reconstruct a more precise real-time radio map. Moreover, the function of MP can be embedded into smartphones or mobile devices, which is considered relatively lowcost to monitor the RSS. Nevertheless, inappropriate deployed locations of MPs cause insufficient information for radio map establishment. Therefore, it is essential to design feasible schemes to cluster the RPs into several groups and to select the cluster head as MPs based on available signal sources and map information [16]. The authors in [17] introduce the affinity propagation clustering algorithm which can deal with the RP clustering problem. The affinity propagation clustering process exchanges similarity by utilizing the responsibility and availability messages among the nodes in each iteration. Note that the responsibility message quantifies how well-suited the node serves as the exemplar to the other nodes; while availability message represents how appropriate the node selects the other node as its exemplar. After the clustering process, the nodes will be divided into several clusters and choose their own cluster head as the exemplar. The advantage of affinity propagation clustering is that the process only requires the similarity among data and is unnecessary to pre-define the number of clusters in most cases. The number of clusters can also be constrained by adjusting specific parameters. Hence, we can choose appropriate MPs from the RPs based on affinity propagation clustering by leveraging a designed similarity function with the aid of useful information.

In this paper, to properly determine the locations of MPs, the proposed RSS-oriented map-assisted clustering (ROMAC) algorithm is designed based on affinity propagation clustering. With the aid of ML-enhanced techniques, cluster-based online database establishment (CODE) is proposed to reconstruct the fingerprint database for solving the time-variation problem without recollecting RSS information of RPs. Moreover, we propose a cluster-scaled location estimation (CsLE) algorithm to further improve the position estimation accuracy by computing adaptive weights based on the RSS's signal variance and cluster information. Unlike conventional WkNN, CsLE can localize the user accurately without selecting the farther RPs as candidates during the online phase. The contributions of this paper are summarized as below.

- We propose a skeleton-assisted learning-based clustering localization (SALC) system that optimizes RP clustering and MP selection in ROMAC, fingerprinting database reconstruction in CODE, and user positioning in CsLE in time-varying indoor environments.
- 2) ROMAC utilizes affinity propagation to cluster RPs associated with the selected MPs as cluster heads. We jointly consider RSS differences, geometric relationship of RPs, and time-variation effect. Note that the clustering information from ROMAC is utilized in both CODE and CsLE for database reconstruction and accurate positioning.
- 3) The proposed CODE algorithm aims for solving the timevariation issue by utilizing linear regression and neural network techniques to generate adaptive database for realtime radio maps. CsLE aims for user positioning based on

ROMAC and CODE, which considers RSS variance from APs. It avoids noisy RSS signals and infeasible faraway RPs as candidate locations.

4) Simulation results show that proposed ROMAC appropriately clusters the RPs as well as chooses the MP cluster head. CODE can efficiently reconstruct a real-time radio map to solve time-variation. While, CsLE reaches a higher localization estimation accuracy than existing methods. Moreover, real-time experiments are conducted to validate the effectiveness of our proposed SALC system, with a higher positioning accuracy.

The remaining of the paper is organized as follows. In Section II, we have investigated the related works. In Section III, the system architecture and flowchart of SALC are demonstrated. The proposed ROMAC, CODE, and CsLE schemes are elaborated in Section IV. In Section V, the simulation and experimental results are discussed in detail. Conclusions are drawn in Section VI.

II. RELATED WORK

Indoor localization has been an interesting research for decades, conventional RSS-based fingerprinting for indoor positioning is studied in [18], [19]. There exist critical bottlenecks restricting its large-scale implementations which consist of time-consuming and labor-intensive data collection process under severe wireless environmental influence, e.g., RSS suffers from dynamic environmental changes such as temperature, shadowing, and obstacles. In [20] and [21], the authors have provided a comprehensive survey of existing fingerprinting methods and dynamic update techniques for radio maps. However, they did not consider the concept of MP deployment, which is capable of efficiently updating the radio map by selecting the appropriate MPs. As a result, online RSS measurement may significantly deviate from the fingerprint database established in the offline phase. However, the RSS will also be affected by time-variation and human blockage, which are not jointly considered in conventional schemes. Furthermore, static fingerprint database may be unreliable, which requires repeated data collection to maintain a satisfactory positioning accuracy. The works of [22]-[24] have conceived different methods to solve the above-mentioned problems. For example, the authors in [22] estimate the RSS at non-site-surveyed positions and utilize the support vector regression (SVR) to improve the resolution of the radio map. Existing researches calibrate the database mainly based on either distance-based path-loss models or interpolation method. However, the paper of [25] has shown that it is difficult to adopt signal loss models in the complex and time-varying environments.

Some existing works intend to modify the classical WkNN to improve localization accuracy by considering different factors [26]–[29]. [26] has proposed a feature-scaled WkNN, depending on the observation of different RSS values and distinguishability in geometrical distances. However, solely taking either RSS or variance [27] into consideration is potentially insufficient to precisely estimate the user position in sophisticated indoor wireless environments. Therefore, the authors in [28] have proposed a new weighted algorithm based



Fig. 1: Network scenario for proposed SALC system. It shows that 3 APs are deployed in two rooms with three clusters labeled with different color, where the corresponding 3 MPs are chosen as cluster heads to perform real-time monitoring of time-varying RSS signals.

on geometric distance of the RPs, and authors in [29] also proposed a restricted WkNN by considering indoor moving constraints to reduce spatial ambiguity. However, the RSS will also be affected by time-variation and human blockage, which is not jointly considered in the conventional schemes. Consequently, it becomes important to investigate the factors disturbing RSS values, which can result in acquiring faraway incorrect target locations with similar RSS.

Under such issues of complex indoor environments and the nonlinearity of radio map caused by time-varying effect, the state-of-the-art machine learning (ML) technique is capable of intelligently estimating user position and of dynamically establishing fingerprint database in an effective and efficient manner. The works in [30], [31] utilize different linear regression methods to calibrate the received signals in order to alleviate the variation of RSS. The works [32]-[34] have proposed linear regression-enhanced methods to update the online radio map based on the observation of real-time received signals. In [35], the authors have applied transfer learning to realize adaptive database construction with the aid of the arbitrarily deployed MPs. The authors in [12] have utilized the federated learning to address dynamic and heterogeneous data streams in indoor localization. In [36], time-varying effect is taken into account with the aid of teacher-student learning. However, it requires laborious data collection as well as sitespecific training. In [12], [36], they both require a much more complex neural network and training mechanism as well as high-dimensional data, which may lead to time-consuming process. Moreover, the existing works did not jointly consider the time-variation and human blockage effects in the online phase. We propose the SALC to address the above-mentioned problem, which will be introduced in the following section in detail.

III. System Architecture and Problem Formulation

Fig. 1 illustrates the WiFi network scenario of indoor localization. In the proposed SALC system, we jointly consider RP clustering, MP deployment, adaptive fingerprinting database reconstruction, and user position prediction based on the RPs' time-varying RSS and geometrical information, which have not been considered in existing literature. The network is deployed with N_{ap} APs, N_{rp} RPs and N_{mp} MPs, where N_{mp} will be determined in the proposed SALC scheme in the next section. During the offline phase of fingerprinting, the measured RSS $\alpha_{n,l}^{RP}(t)$ is received from the l^{th} AP on the n^{th} RP at the t^{th} time instant, where the location of the corresponding RP is also recorded. The offline database containing collected RSS measurements from all RPs is represented as

$$\boldsymbol{\alpha}_{l}^{RP}(t) = [\alpha_{1,l}^{RP}(t), \dots, \alpha_{n,l}^{RP}(t), \dots, \alpha_{N_{rp},l}^{RP}(t)]^{\top}, \quad (1)$$

where $l \in [1, N_{ap}]$ and $n \in [1, N_{rp}]$ are the indexes of APs and RPs, respectively, and \top is defined as transpose operation. Note that $\alpha_{n,l}^{RP}(t)$ is estimated based on the path-loss model specified in [37], which is expressed as

$$\alpha_{n,l}^{RP}(t) = P_t - 20\log(f) + P_d\log(d_{n,l}) - 28, \qquad (2)$$

where P_t is the transmit power, f is the operating frequency, P_d is the path-loss coefficient depending on indoor environments, and $d_{n,l}$ is the distance between the n^{th} RP and the l^{th} AP. Note that $\alpha_{n,l}^{RP}(t)$ is in the unit of dB. The locations of MPs will be determined among RPs as cluster heads considering the similarity among RPs with respect to time, RSS, and geometric distance, which aim for monitoring the timevarying RSS at time t in its own cluster. The measurement of MP's RSS can be given by

$$\boldsymbol{\alpha}_{l}^{MP}(t) = [\alpha_{1,l}^{MP}(t), \dots, \alpha_{m,l}^{MP}(t), \dots, \alpha_{N_{mp},l}^{MP}(t)]^{\top}, \quad (3)$$

where $m \in [1, N_{mp}]$ is the index of MPs and $\alpha_{m,l}^{MP}(t)$ is estimated based on the path-loss model the same as that in (2). In the online phase, the adaptive fingerprinting database will be generated by applying both real-time received RSS from MPs and offline established RSS database. Therefore, the user's position can be estimated by matching its real-time RSS $\alpha_l^U(t)$ to that in the generated fingerprinting database.

The flowchart of the proposed SALC system is shown in Fig. 2, which is composed of three sub-algorithms, including ROMAC, CODE, and CsLE. The main target of SALC is to deal with the time-variation issues in conventional fingerprinting database resulting in incorrect radio map matching, which consequently incurs a lower location estimation accuracy. By monitoring the environmental changes with the aid of MPs, the distribution of $\alpha_{n,l}^{RP}(t)$ can be well-estimated to establish the adaptive RP database as

$$p\left(\alpha_{n,l}^{RP}(t_o)\right) = p\left(\alpha_{n,l}^{RP}(t)|\alpha_{m,l}^{MP}(t_o)\right),\tag{4}$$

where t_o indicates the measurement time step in online phase. We need to obtain the RP's RSS value $\alpha_{n,l}^{RP}(t)$ that maximizes the distribution $p(\alpha_{n,l}^{RP}(t_o))$ on each RP. Therefore, based on (4), we can acquire more precise RSS on each RP which is formulated as

$$\boldsymbol{\alpha}_{l}^{RP}(t_{o}) = M\left(\boldsymbol{\alpha}_{l}^{MP}(t_{o})\right) = \operatorname*{argmax}_{\alpha_{n,l}^{RP}(t)} p\left(\alpha_{n,l}^{RP}(t) | \alpha_{m,l}^{MP}(t_{o})\right),$$
(5)

where $M(\alpha_l^{MP}(t_o)) : \mathbb{R}^{N_{mp}} \to \mathbb{R}^{N_{rp}}$ indicates the mapping function from MPs to RPs. We can observe from (5) that



Fig. 2: System flowchart of proposed skeleton-assisted learning-based clustering localization (SALC) which includes ROMAC, CODE, and CsLE dealing with RPs clustering, MPs selection, fingerprinting database reconstruction, and user positioning.

it leads to a non-linear optimization problem due to the sophisticated indoor environment with ample multi-paths and blockages. The proposed ROMAC in the upper-left of Fig. 2 is adopted to select the MPs in (5) to provide instant information of $\alpha_{m,l}^{MP}(t_o)$. Furthermore, the proposed CODE in the lower-left of Fig. 2 is designed to generate the distribution of $\alpha_{n,l}^{RP}(t_o)$ by employing deep neural networks in order to solve the non-linear estimation problem in (5). We can then obtain the estimated user position $\hat{x}(t_o) = [\hat{x}(t_o), \hat{y}(t_o)]^{\top}$ according to the received RSS $\alpha^U(t_o)$ at its real position $x = [x, y]^{\top}$, which is expressed as

$$\hat{\boldsymbol{x}}(t_o) = \operatorname*{argmax}_{\forall \boldsymbol{x}} p\left(\boldsymbol{x} | \boldsymbol{\alpha}^U(t_o)\right), \tag{6}$$

where $\alpha^U(t_o) = [\alpha_1^U(t_o), \dots, \alpha_l^U(t_o), \dots, \alpha_{N_{ap}}^U(t_o)]$. The localization problem (6) is equivalent to the following formula as

$$p\left(\boldsymbol{x}|\boldsymbol{\alpha}^{U}(t_{o})\right) = \frac{p\left(\boldsymbol{\alpha}^{U}(t_{o})|\boldsymbol{x}\right)p\left(\boldsymbol{x}\right)}{p\left(\boldsymbol{\alpha}^{U}(t_{o})\right)},$$
(7)

where the posterior distribution $p(\boldsymbol{x}|\boldsymbol{\alpha}^{U}(t_{o}))$ can be derived based on Bayes' theorem [38]. The probability $p(\boldsymbol{x})$ follows the uniform distribution, which therefore can be ignored due to its proportionality property. Moreover, the distribution of RSS $\boldsymbol{\alpha}^{U}(t_{o})$ is attainable because RSS is collected from the mobile device. Accordingly, maximizing likelihood $p(\boldsymbol{\alpha}^{U}(t_{o})|\boldsymbol{x})$ is equivalent to maximizing the radio map mapping function $R(\boldsymbol{x}): \mathbb{R}^{N_{rp}} \to \mathbb{R}^{2}$ which can be represented as

$$R(\boldsymbol{x}) = \operatorname*{argmax}_{\boldsymbol{\alpha}^{U}(t)} p\left(\boldsymbol{\alpha}^{U}(t)|\boldsymbol{x}\right) \approx \sum_{n=1}^{N_{rp}} \alpha_{n,l}^{RP}(t)\delta(\boldsymbol{x}-\boldsymbol{x}_{n}), \quad (8)$$

where $\delta(\cdot)$ is a delta function with value equal to 1 if $\boldsymbol{x} = \boldsymbol{x}_n$, and \boldsymbol{x}_n is the location of the n^{th} RP. The radio map $R(\boldsymbol{x})$ is approximated from the summation of updated parameter $\alpha_{n,l}^{RP}(t)$ from (4). 4

IV. PROPOSED SALC SYSTEM

The proposed SALC system provides location estimation with the employment of fingerprinting and received RSS values from MPs. However, the data collection will suffer from the non-linear time-variation issue during both offline and online phases. To deal with this issue, it becomes crucial to design a non-linear MP-aided mapping function in order to reconstruct the real-time radio map. Therefore, we propose three sub-algorithms including ROMAC, CODE, and CsLE in the SALC system to solve the above-mentioned issues, i.e., RPs clustering, MPs selection, fingerprinting database reconstruction, and user positioning. The ROMAC is mainly designed to divide the RPs into clusters and select corresponding MPs, whereas CODE is adopted to generate the distribution of $\alpha_l^{RP}(t)$ in order to solve the non-linear problem in (5). The proposed CsLE scheme aims for estimating user's position according to the adaptive database constructed by CODE and the clustering information obtained from ROMAC.

A. RSS-Oriented Map-Assisted Clustering (ROMAC)

The proposed ROMAC scheme is designed based on an unsupervised learning approach, which can classify the unlabeled data into different groups based on attainable RSS features. As mentioned in Section II, the existing methods for the MP deployment have not jointly considered the signal strength, map information and time-varying effect. In this subsection, we will introduce how ROMAC jointly considers all important factors, including RSS measurements, skeletonbased shortest path (SSP), and time-variation characteristics to conduct the clustering process. Furthermore, deployment of MPs is also determined by selecting the cluster head. ROMAC is designed based on the affinity propagation [39], which only requires the similarity feature among RPs, without the need to pre-define the number of clusters. The self-defined similarity consists of three key factors including the amplitude difference of RSS among RPs, layout of RPs, and time-variance of RSS. The amplitude difference of RSS represents potential signal fading and blockages. The indoor layout provides the position knowledge for the SSP, whilst the time-variance of RSS reflects the time-varying effect of signals. Based on the observed received signal $\alpha_l^{RP}(t)$, the difference of RSS between the i^{th} and j^{th} RPs is defined as

$$d_{RSS}(i,j) = \frac{1}{N} \sum_{k=1}^{N} \left| \alpha_{i,l}^{RP}(t_k) - \alpha_{j,l}^{RP}(t_k) \right|, \qquad (9)$$

where N is the considered time interval, t_k is the time index for $k \in [1, N]$ and the notion of |x| represents the absolute value of x. Notice that the difference between the i^{th} and j^{th} RP's RSS is related to its Euclidean distance, where a smaller value of $d_{RSS}(i, j)$ represents a higher similarity level between RPs.

Fig. 3 shows an exemplified layout of an indoor environment, where the white area is walkable area and the black lines are the walls. The red points are RPs to be estimated and blue lines and vertices are formed by the SSP scheme adopted from [40] to acquire the spatial skeleton, which provides a compact



Fig. 3: The schematic diagram of spatial skeleton and RPs. The white area is walkable area and the black lines are the walls. The red points in walkable area are RPs, the blue lines represent the skeleton and the blue points are the vertices of skeleton. The shortest path $d_{SSP}(i, j)$ between the i^{th} and the j^{th} RPs is also illustrated.

map of the shortest paths among RPs. The vertices of SSPskeletons are calculated based on generalized Voronoi diagram (GVD) technique [41] as an enhancement of conventional Voronoi diagram in order to partition a plane into several cell-like regions. The output of GVD contains edges and vertices as demonstrated in blue lines and points in Fig. 3, respectively. We define D as an SSP matrix which is given by $D = [d_{v,w}], \quad \forall v, w \in [1, N_v]$, where $d_{v,w}$ is the shortest path from the v^{th} to w^{th} vertices, and N_v is the total number of vertices. Therefore, as shown in Fig. 3, the spatial distance of RPs can be derived from the shortest path between the i^{th} and the j^{th} RPs as

$$d_{SSP}(i,j) = d(g_i, v_i) + d_{i,j} + d(g_j, v_j),$$
(10)

where $i, j \in [1, N_{rp}]$ are the indexes of RPs, v_i and v_j are respectively the nearest vertex of the i^{th} and the j^{th} RP. g_i and g_j are the positions of i^{th} and j^{th} RPs, respectively. $d(g_i, v_i)$ denotes the Euclidean distance between i^{th} RP and vertex v_i , whilst $d_{i,j}$ denotes the shortest path between the i^{th} and j^{th} vertices. Therefore, $d_{SSP}(i, j)$ reflects the spatial relationship between the i^{th} and j^{th} RPs in an SSP-aided layout. Note that smaller value of $d_{SSP}(i, j)$ indicates higher spatial relationship.

Moreover, the long-term difference of RSS time-varying between empty and crowded areas is considered. The difference between the i^{th} and the j^{th} RP is derived from

$$\delta_{i,j} = \left| \sum_{l=1}^{N_{ap}} \sum_{k=1}^{N} [\alpha_{i,l}^{RP}(t_k) - \alpha_{i,l}^{RP}(t_e)] - [\alpha_{j,l}^{RP}(t_k) - \alpha_{j,l}^{RP}(t_e)] \right|^{-1},$$
(11)

where $\alpha_{i,l}(t_e)$ is RSS measured on the i^{th} RP from l detectable APs which is considered as a reference RSS obtained at the time instant t_e , e.g., RSS acquired from an empty area. The parameter $\delta_{i,j}$ reflects the distinct difference of time-varying effect between the i^{th} and j^{th} RPs. Notice that a higher value of $\delta_{i,j}$ indicates that the RPs are less susceptible to time-variation effect. For example, consider the case that $\delta_{1,2} = 1/(11-1) = 0.1$ with $[\alpha_{1,l}^{RP}(t_k) - \alpha_{1,l}^{RP}(t_e)] = 11$ and $[\alpha_{2,l}^{RP}(t_k) - \alpha_{2,l}^{RP}(t_e)] = 1$; while $\delta_{3,4} = 1/(2-1) = 1$ with

 $[\alpha_{3,l}^{RP}(t_k) - \alpha_{3,l}^{RP}(t_e)] = 2$ and $[\alpha_{4,l}^{RP}(t_k) - \alpha_{4,l}^{RP}(t_e)] = 1$. It can be intuitively observed that the area around RPs 1 and 2 with $\delta_{1,2} = 0.1$ is more susceptible to time-varying effect, e.g., a crowded room, comparing to that for RPs 3 and 4 with $\delta_{3,4} = 1$, e.g., an empty room. The RPs that are highly influenced by time-variation, i.e., smaller δ_i, j will be treated with larger similarity, since the proposed MPs are designed to resist time-varying effect.

According to RSS, SSP and the time-varying effect in (9), (10) and (11), respectively, the joint similarity $s_{i,j}$ between the i^{th} and the j^{th} RPs can be formulated, which is also shown at top-left of Fig. 2, as

$$s_{i,j} = -\boldsymbol{\omega} \cdot \left[d_{RSS}(i,j), d_{SSP}(i,j), \delta_{i,j} \right]^{\top}, \qquad (12)$$

where $\boldsymbol{\omega} = [\omega_{RSS}, \omega_{SSP}, \omega_{\delta}]$ represents the important weights of RSS, SSP and time-varying effect. Notice that we impose the negative sign on the factors in (12) to reflect smaller values of those three factors resulting in larger joint similarity. We denote S_j as the joint similarity between the j^{th} RP and the others, which is defined as

$$S_j = [s_{1,j}, \dots s_{i,j}, \dots, s_{N_{rp},j}],$$
 (13)

where $\forall i \neq j$, and $i, j \in [1, N_r p]$. Furthermore, we define the preference to represent the self-similarity of the j^{th} RP as

$$s_{j,j} = M_d(\boldsymbol{S}_j),\tag{14}$$

where $M_d(\cdot)$ is the median function averaging all elements of a matrix. The preference indicates the probability of a specific RP becoming a cluster exemplar, i.e., it is selected as the corresponding MP. Accordingly, a lower preference value of an RP indicates that it behaves similar to the other RPs, which means that it possesses a lower chance to be selected as a cluster exemplar.

Based on the SSP-aided map and similarity definition, the proposed ROMAC considers two types of messages among RPs including responsibility message $r_{i,j}$ and availability message $a_{i,j} \forall i, j \in [1, N_{rp}]$. The message will be exchanged iteratively in order to derive the prioritized responsibility and availability messages. The responsibility $r_{i,j}$ is sent from the i^{th} RP to the j^{th} RP, which is defined as

$$r_{i,j} = s_{i,j} - \max_{j' \neq j} \{ a_{i,j'} + s_{i,j'} \},$$
(15)

where $\max{\{\cdot\}}$ function gives the maximum value among input elements. The availability $a_{i,j}$ is then sent from the j^{th} RP to the i^{th} RP represented by

$$a_{i,j} = \begin{cases} \min\{0, r_{j,j} + \sum_{i' \neq i,j} \max\{0, r_{i',j}\}\}, & \forall i \neq j, \\ \sum_{i' \neq j} \max\{0, r_{i',j}\}, & \forall i = j, \end{cases}$$
(16)

where $\max\{\cdot, \cdot\}$ will choose a larger element as the output and $\min\{\cdot, \cdot\}$ will choose a smaller one. (15) and (16) show that higher $r_{i,j}$ representing that the j^{th} RP is more appropriate to serve as the candidate exemplar for the i^{th} RP, whereas higher $a_{i,j}$ indicates that the i^{th} RP has higher tendency to select the j^{th} RP as its exemplar. The exemplar will be determined after both responsibility $r_{i,j}$ and availability $a_{i,j}$ matrix are updated. Consequently, the j^{th} RP with the maximum value

Algorithm 1: RSS-oriented map-assisted clustering (ROMAC)

Input : Similarity $s_{i,j} = -\boldsymbol{\omega} \cdot [d_{RSS}(i,j), d_{SSP}(i,j), \delta_{i,j}]^{\top}$ **Output:** Exemplar set Ψ , cluster exemplar E(i), cluster set C(m)1 Initialize responsibility $r_{i,j}$ and availability $a_{i,j}$ to 0 while the cluster set C(m) is changed do 2 Update responsibilities: 3 $\begin{array}{l} r_{i,j} \leftarrow s_{i,j} - \max_{j' \neq j} \{a_{i,j'} + s_{i,j'}\} \\ \text{Update availabilities: } a_{i,j} \leftarrow \end{array}$ 4 $\begin{cases} \min\{0, r_{j,j} + \sum_{i' \neq i,j} \max\{0, r_{i',j}\}\} \\ \sum_{i' \neq j} \max\{0, r_{i',j}\} \end{cases}$ $\forall i \neq j,$ $\forall i = j,$ Update exemplar set: 5 $\Psi \leftarrow \{\mu | \mu = \operatorname{argmax}_{\forall i,j \in [1,N_{rp}]} \{r_{i,j} + a_{i,j}\}\}$ Select MP for *i*th RP from exemplar set: 6 $E(i) \leftarrow \operatorname{argmax}_{i \in \Psi} \{ r_{i,j} + a_{i,j} \}$ Update cluster set: $C(m) \leftarrow \{i | E(i) = \mu\}$ 7

of $(r_{i,j} + a_{i,j})$ will be selected as the cluster exemplar MP. The set of exemplars Ψ can be derived as

$$\Psi = \{\mu | \mu = \operatorname*{argmax}_{\forall i, j \in [1, N_{rp}]} \{r_{i, j} + a_{i, j}\}\}.$$
(17)

Notice that the number of MPs can therefore be determined as $N_{mp} = \operatorname{rank}(\Psi)$ according to the proposed ROMAC scheme. Furthermore, for the i^{th} RP, its exemplar MP is selected from the exemplar set Ψ as

$$E(i) = \operatorname*{argmax}_{j \in \Psi} \{r_{i,j} + a_{i,j}\}.$$
(18)

Alternatively, the m^{th} RP as the i^{th} RPs' exemplar MP is defined as the set of

$$C(m) = \{i | E(i) = \mu\},$$
(19)

where $i \in [1, N_{rp}]$ is the index of RP in the m^{th} cluster. Note that the parameter μ denotes the selected exemplar MP's actual RP's index number, whilst m is defined as the re-ordered index of μ for the ease of representation in the following design. For example, consider the case that $\Psi = \{\mu | \mu = 6, 14, 22, 34\}$, the first cluster's exemplar MP with m = 1 becomes C(1) = $\{i | E(i) = 6\}$; while the second MP with m = 2 is C(2) = $\{i | E(i) = 14\}$. The iterations will be executed until the cluster set C(m) becomes unchanged. As illustrated in Fig. 2, both the MP set Ψ in (17) and the cluster set C(m) in (19) will be utilized as the inputs of the following CODE scheme. The concrete procedure of ROMAC is provided in Algorithm 1.

B. Cluster-based Online Database Establishment (CODE)

In the proposed CODE scheme, both the linear regression (CODE-LR) and neural network (CODE-NN) schemes are designed in Fig. 2. CODE has addressed expired fingerprinting database issue caused by the time-varying effects. The radio map information can be predicted by adopting either regression or neural network models acting as the database generator.

The proposed CODE-LR scheme acquires the distribution in (4) considering the linearity property of signal strength, i.e., RSS is approximately inversely proportional to the distance between the transmitter and receiver in a linear manner. To further account for nonlinear effects caused by indoor signal fading or moving objects, the conceived CODE-NN scheme improves the accuracy of database prediction by extracting latent information from a deep neural network model.

1) CODE-LR: The proposed CODE-LR algorithm is designed based on SVR [42] to conduct online database construction. We intend to develop a predictive modeling technique investigating the relationship between dependent and independent variables to represent the measured RSS on MPs and RPs, respectively. The regression models of RPs will be trained considering the long-term RSS information in the offline database construction phase. During the online phase, the coefficients of regression models are extracted to establish the fingerprint database for online matching process. Note that the overhead of time-consuming offline database construction can therefore be reduced with the assistance of online adjustment to maintain satisfactory positioning accuracy of SALC.

Under time-varying environment, the RSS of each RP varies individually when environment changes. However, with the aid of cluster information C(m) and deployed MPs Ψ obtained from ROMAC, we are capable of observing real-time RSS value. We notice that RPs in certain cluster behave similarly to their corresponding cluster exemplar MP, where the similarity pattern can be acquired via proposed CODE-LR. Considering that RP $n \in C(m)$ and MP $m \in \Psi$ represent the cluster head of RPs $n \in C(m)$, the estimated RSS $\hat{\alpha}_{n,l}^{RP}(t_k)$ from the l^{th} AP can be calculated by

$$\hat{\alpha}_{n,l}^{RP}(t_k) = c_{n,l}^R \alpha_{m,l}^{MP}(t_k) + b_{n,l}^R, \qquad (20)$$

where $c_{n,l}^R$ and $b_{n,l}^R$ are coefficient and bias, respectively, of regression in CODE-LR. The loss function is modeled as

$$J_{n,l} = \frac{1}{2} \sum_{k=1}^{N} \left(\hat{\alpha}_{n,l}^{RP}(t_k) - \alpha_{n,l}^{RP}(t_k) \right)^2.$$
(21)

To iteratively update the weights of regression, the stochastic gradient descent is adopted as

$$b_{n,l}^{R} = b_{n,l}^{R} - \eta \frac{\partial J_{n,l}}{\partial b_{n,l}^{R}} = b_{n,l}^{R} - \eta \left[\hat{\alpha}_{n,l}^{RP}(t_k) - \alpha_{n,l}^{RP}(t_k) \right],$$
(22)

$$c_{n,l}^{R} = c_{n,l}^{R} - \eta \frac{\partial J_{n,l}}{\partial c_{n,l}^{R}}$$
$$= c_{n,l}^{R} - \eta \left[\alpha_{m,l}^{MP}(t_{k}) \cdot \left(\hat{\alpha}_{n,l}^{RP}(t_{k}) - \alpha_{n,l}^{RP}(t_{k}) \right) \right], \quad (23)$$

where η is the learning rate. The iteration will execute until the coefficient remains unchanged. Notice that we build a regression model for every RP, and the complete coefficient and bias sets can be represented as $c^R = \{c_{n,l}^R \mid \forall n \in [1, N_{rp}], \forall l \in [1, N_{ap}]\}$ and $b^R = \{b_{n,l}^R \mid \forall n \in [1, N_{rp}], \forall l \in [1, N_{ap}]\}$. These two parameter sets will be obtained from the model training process of CODE-LR, as shown in the bottom-left part of Fig. 2 and saved in the offline model parameter database.



Fig. 4: Network model for proposed CODE-NN scheme.

Furthermore, during the online phase, the online database generation will compute real-time RSS for RPs $\hat{\alpha}_{n,l}^{RP}(t_o)$ as shown in the bottom-right part of Fig. 2 as

$$\hat{\alpha}_{n,l}^{RP}(t_o) = c_{n,l}^R \alpha_{m,l}^{MP}(t_o) + b_{n,l}^R,$$
(24)

where t_o denotes the time instant at online phase. $c_{n,l}^R$ and $b_{n,l}^R$ are respectively acquired from c^R and b^R in model parameter database at the offline stage. Consequently, the RSS values from all RPs $\hat{\alpha}_{n,l}^{RP}(t_o)$ are computed and stored in the adaptive RP database in order to reconstruct the real-time radio map, which will be utilized for location estimation.

2) CODE-NN: The linear relationship between the RSS values of MPs and RPs considered in CODE-LR may be impractical in realistic system due to complicated indoor wireless environments. Therefore, CODE-NN scheme is proposed to perform nonlinear training and mapping. The training process is divided into pre-training and fine-tuning phases. In the pre-training phase, the loss function and back-propagation will be computed by a cluster-averaged target in order to take cluster information into account to represent various similarity levels between RPs. In the fine-tuning phase, the pre-trained parameters will be loaded as initial parameters associating with the unaveraged target function in order to guarantee feasible prediction results.

First of all, data preprocessing as shown in CODE-NN block of Fig. 2 is applied in order to consider the time-varying effects. With the same cluster's RP $n \in C(m)$ and MP $m \in \Psi$, we subtract RSS from the l^{th} AP in the initial environment $\alpha_{n,l}^{RP}(t_e)$ from that at time t_k , i.e., $\alpha_{n,l}^{RP}(t_k)$, to obtain the time-variant difference as

$$\delta_{n,l}^{RP}(t_k) = \alpha_{n,l}^{RP}(t_k) - \alpha_{n,l}^{RP}(t_e).$$
 (25)

Similarly, the difference of time-varying RSS obtained at the m^{th} MP $\delta_{m,l}^{MP}(t_k)$ can be defined as

$$\delta_{m,l}^{MP}(t_k) = \alpha_{m,l}^{MP}(t_k) - \alpha_{m,l}^{MP}(t_e).$$
 (26)

Fig. 4 shows the network architecture of CODE-NN including the pre-training and fine-tuning phases. The network is constructed by the input and output layers with dimensions N_{mp} and N_{rp} , respectively. Five fully connected neural network (FCN) layers are chosen as the hidden layers with the corresponding number of network nodes. Notice that the RSS difference $\delta_{m,l}^{MP}(t_k)$ acquired from all N_{mp} MPs will be served as the inputs to FCN to provide the learning mechanism

with an integrated fashion. The model output $\hat{\delta}_{n,l}^{RP}(t_k)$ can be computed by the forward propagation from hidden layers shown as the green part in Fig. 4, which is represented as

$$\hat{\delta}_{n,l}^{RP} = f^{(5)} \left(\sum_{m=1}^{D_5} \dots f^{(1)} \left(\sum_{m=1}^{D_1} w_{n,m}^{(1)} \delta_{m,l}^{MP} + b_n^{(1)} \right) + b_n^{(5)} \right),$$
(27)

where $f^{(h)}(\cdot)$ is the activation function and D_h is the number of nodes of the h^{th} hidden layer for h = 1 to h = 5. The dimension of hidden layer's weight $w_{n,m}^{(h)}$ is $D_h \times D_{h-1}$ for h = 2 to 5 and $D_1 \times N_{mp}$ for h = 1, whilst that for the bias $b_n^{(h)}$ is $D_h \times 1$ for all h.

In the pre-training phase, the goal is to provide initial parameter of every node, which can help the model find the solution rapidly, i.e., to reduce the iterations of training process. After the data preprocessing in Fig. 2, the RSS difference $\delta_{n,l}^{RP}(t_k)$ at the k^{th} time instant from (27) are averaged within each cluster since the RPs in the same cluster have similar trend in time-variation. With the aid of cluster information Ψ and C(m), the averaged target can be represented as

$$\bar{\delta}_{n,l}^{RP}(t_k) = \frac{1}{N_m} \sum_{n=1}^{N_m} \delta_{n,l}^{RP}(t_k),$$
(28)

where N_m is the number of RPs in the m^{th} cluster and n is the index of RP, $\forall n \in C(m)$. Compared to utilizing $\delta_{n,l}^{RP}(t_k)$, choosing $\bar{\delta}_{m,l}^{RP}(t_k)$ in (28) as the target in pre-training phase can reduce computational complexity of loss function. Therefore, the loss function is designed between the target $\bar{\delta}_{n,l}^{RP}(t_k)$ and the output $\hat{\delta}_{n,l}^{RP}(\theta_p^{\tau};t_k)$ as

$$L_{\gamma}\left(\bar{\delta}_{n,l}^{RP}(t_{k}),\hat{\delta}_{n,l}^{RP}(\boldsymbol{\theta}_{p}^{\tau};t_{k})\right) = \begin{cases} \frac{1}{2} \left[\bar{\delta}_{n,l}^{RP}(t_{k}) - \hat{\delta}_{n,l}^{RP}(\boldsymbol{\theta}_{p}^{\tau};t_{k})\right]^{2}, \text{ for } \left|\bar{\delta}_{n,l}^{RP}(t_{k}) - \hat{\delta}_{n,l}^{RP}(\boldsymbol{\theta}_{p}^{\tau};t_{k})\right| \leq \gamma, \\ \gamma \left|\bar{\delta}_{n,l}^{RP}(t_{k}) - \hat{\delta}_{n,l}^{RP}(\boldsymbol{\theta}_{p}^{\tau};t_{k}) - \frac{1}{2}\gamma^{2}\right|, \text{ otherwise,} \end{cases}$$

$$(29)$$

where $\hat{\delta}_{n,l}^{RP}(\boldsymbol{\theta}_p^{\tau}; t_k)$ denotes the model output predicted by the parameter at the τ^{th} training iteration during the pre-training phase. We select Huber loss [43] as the loss function in order to eliminate the effects of outliers by setting the threshold γ . The gradient descent method is utilized to search the optimum of the loss function during back propagation as shown in Fig. 4. The gradient can be updated iteratively as

$$\boldsymbol{\theta}_{p}^{\tau+1} = \boldsymbol{\theta}_{p}^{\tau} - \eta \cdot \nabla L_{\gamma} \left(\bar{\delta}_{n,l}^{RP}(t_{k}), \hat{\delta}_{n,l}^{RP}(\boldsymbol{\theta}_{p}^{\tau}; t_{k}) \right), \qquad (30)$$

where θ_p^{τ} is the parameter including the weight and bias during the pre-training phase, η is learning rate, and $\nabla L_{\gamma}(\cdot, \cdot)$ is the first-order derivative of the loss function.

After the pre-training phase, the following fine-tuning phase will be performed as shown in both Figs. 2 and 4. The parameter θ_p will be saved and treated as the initial parameter for fine-tuning process, i.e., $\theta_f^0 = \theta_p^T$, where θ_f^0 and θ_p^T represent the initial network parameter during fine-tuning phase and the network parameter updated at the last iteration T in pre-training phase, respectively. Notice that the architecture of neural network and the number of nodes in fine-tuning

phase are designed to be the same as those in the pre-training phase, where the loss function in fine-tuning can be acquired by replacing the averaged target $\bar{\delta}_{n,l}^{RP}(t_k)$ in (29) with $\delta_{n,l}^{RP}(t_k)$ for each RP. Note that we also update the weights and bias of neural networks via the gradient descent method as that in (30) by substitute θ_p^{τ} with θ_f^{τ} . With the initial parameter acquired from the pre-training phase, the gradient descent can converge faster to find the optimum solution.

After completion of pre-training and fine-tuning phases, the parameter θ_f will be saved into the model parameter database to reconstruct the real-time radio map $\hat{\alpha}_{n,l}^{RP}(t_o)$ at the online phase. As shown in the lower-right of Fig. 2, the time-variation $\hat{\delta}_{n,l}^{RP}(t_o)$ is predicted based on the online monitored MP's RSS difference $\delta_{m,l}^{MP}(t_o)$ and the trained model parameter θ_f via (27). Once the network completes the calculation in online phase and outputs the prediction of $\hat{\delta}_{n,l}^{RP}(t_o)$, the element stored in the adaptive RP database in the online phase can be represented as

$$\hat{\alpha}_{n,l}^{RP}(t_o) = \hat{\delta}_{n,l}^{RP}(t_o) + \alpha_{n,l}^{RP}(t_e).$$
(31)

Note that $\hat{\alpha}_{n,l}^{RP}(t_o)$ represents the real-time adaptive radio map, which will be used to estimate the user position by the following CsLE algorithm. Different from the linear property of CODE-LR, CODE-NN can solve the nonlinear problem of radio signal propagation such as human signal-blocking based on model training and adaptation.

C. Cluster-scaled Location Estimation (CsLE)

In order to accurately estimate the user position in the online phase, the proposed CsLE scheme takes into account the signal variance caused by time-varying effects. As shown in the top-right part of Fig. 2, CsLE is implemented based on real-time RSS collected from user $\alpha_l^U(t_o)$, cluster information with RP $n \in C(m)$ acquired from ROMAC, and the adaptive RSS value $\hat{\alpha}_{n,l}^{RP}(t_o)$ for the n^{th} RP via CODE. The modified Euclidean distance (MED) of online received RSS values from the l^{th} AP between the n^{th} RP and user is derived as

$$d_{n,l}(t_o) = \sum_{m=0}^{M} \frac{\hat{\alpha}_{n,l}^{RP}(t_o) - \alpha_l^U(t_o)}{\sigma_m \left(\hat{\alpha}_{n,l}^{RP}(t_o)\right)},$$
(32)

where $\sigma_m(\cdot)$ is the weight scaling function that gives realtime estimated standard deviation of RSS from the n^{th} RP within m^{th} cluster. Higher value of $\sigma_m(\hat{\alpha}_{n,l}^{RP}(t_o))$ indicates that the n^{th} RP's RSS from the l^{th} AP is less reliable among all RPs within C(m). Consequently, the corresponding weight $w_{n,l}(t_o)$ can be chosen as the inverse of estimated MED of each RP n as

$$w_{n,l}(t_o) = \frac{1}{d_{n,l}(t_o)}.$$
(33)

The total set of weights can be defined in a sorted set as $w_l = \{w_{n,l} | w_{n,l} > w_{n',l}, n < n' \forall n, n' \in C(m)\}$, whilst we select the indexes with the first k largest weights to be $w_{k,l} = \{w_{1,l}, \ldots, w_{n,l}, \ldots, w_{k,l} | n \in w_l\}$. Therefore, the estimated

Parameters of system	Value
Number of WiFi APs	3
Number of RPs	56
Number of TPs	89
Indoor topology	$8 \times 10 m^2$
Size of each grid	$1.2 \times 1.2 \ m^2$
Carrier frequency f	2.4 GHz
Channel bandwidth	22 MHz
Number of sample per RP	10 samples
Number of nodes in hidden layers	$\{64, 256, 512, 128, 64\}$
Learning rate η	0.1
Threshold γ	1
Number of nearest neighbor k	3
Pre-training data (sim./exp.)	$\{1120, 4200\}$ samples
Training data (sim./exp.)	$\{2240, 8400\}$ samples

user position $\hat{x}(t_o) = [\hat{x}(t_o), \hat{y}(t_o)]$ acquired at time instant t_o can be computed by

$$\hat{\boldsymbol{x}}(t_o) = \frac{\sum_{l \in N_{ap}} \sum_{w_{n,l} \in \boldsymbol{w}_{k,l}} w_{n,l} \cdot \boldsymbol{x}_n}{N_{ap} \cdot \sum_{w_{n,l} \in \boldsymbol{w}_{k,l}} w_{n,l}}, \qquad (34)$$

where x_n is the geometric location of the n^{th} RP. The proposed CsLE scheme incorporates both cluster information from ROMAC and the RSS relationship between the user and RPs. By leveraging the cluster information, RPs having similar RSS values but located farther away are excluded from the top k^{th} weighting elements, which enhances the accuracy of location estimation.

V. PERFORMANCE EVALUATION

A. Simulation Results

We firstly evaluate the performances of proposed SALC system including ROMAC, CODE and CsLE schemes via simulations. We employ Wireless InSite which is a widelyadopted simulation software to emulate ray-tracing based indoor wireless propagations. We consider a two-room scenario with each room size of $8 \times 10 \text{ m}^2$ as shown in Fig. 5(a), where three APs (marked as large green squares) are deployed at the room corners operating at the frequency of 2.4 GHz. There are 56 RPs (marked as small red squares) evenly distributed with a inter-RP distance¹ of 1.2 m, whilst 89 random test points (TPs) are set in both rooms. As depicted in Fig. 5(a), two different cases are considered as follows: (a) both rooms are empty and (b) left part of top room is crowded with people and bottom room is vacant. We sample 10 time slots to generate different channel conditions for each RP, and the weights in (12) is chosen as $\omega = [1/3, 1/3, 1/3]$. Furthermore, the number of nodes in hidden layers of FCN are designed as $\{64, 256, 512, 128, 64\}$ as shown in Fig. 4. The learning rate η and threshold γ in CODE-NN is set as 0.1 and 1, respectively. The rectified linear unit (ReLU) is selected as the activation

¹In our proposed system, the RPs are placed in a pattern of uniform grids. We empirically find in our several experiment trials that the RP location will not substantially affect the performance of the proposed scheme. The deployment of RPs can potentially enhance the performance, which is proved in some existing works. However, the optmal RP deployment requires a much more complex scheme, which is out of the scope of this paper and can be left as the future work.

TABLE II: Computational Complexity Comparison

Scheme	Computational Complexity
CSE [22]	$\mathcal{O}\left(N^3 \times N_{rp} \times N_{ap} \times \kappa\right)$
CODE-LR	$\mathcal{O}\left(N_{rp} \times N_{ap}\right)$
CODE-NN	$\mathcal{O}\left(N \times N_{mp} \times \prod_{i=1}^{N_L} K_i^2 \times N_{rp} \times N_{ap}\right)$



Fig. 5: (a) Network setup with two-room layout for simulations. (b) The generated skeletons from ROMAC of proposed SALC system. The clustering results by considering similarity of (c) the difference of RSS $d_{RSS}(i, j)$ in (9); (d) the shortest path $d_{SSP}(i, j)$ in (10); (e) the time-varying RSS difference $\delta_{i,j}$ in (11); and (f) the combined factors $s_{i,j}$ in (12) of ROMAC.

function $f^{(h)}(\cdot)$ in (27). The number of nearest neighbor is set to k = 3 in CsLE scheme. The volume of pre-training and training data is 1120 and 2240 samples, which corresponds to 20 and 40 samples per RP, respectively. Table I summarizes the parameter setting in simulations.

Table II elaborates the computational complexity of CODE-LR and CODE-NN compared to the existing method of CSE [22]. CSE has the highest complexity order of $\mathcal{O}(N^3 \times N_{rp} \times N_{ap} \times \kappa)$, where N indicates the data size and κ stands for the kernel size. Note that N^3 comes from the cross-comparison of input in support vector regression mechanism, whereas κ will depend on what kind of kernel is adopted for cross-comparison. The proposed CODE-LR scheme has the lowest computational complexity order of $\mathcal{O}(N_{rp} \times N_{ap})$, which is only proportional to the network deployment size N_{ap} as well as measuring points N_{rp} . Since the input feature of CODE-LR requires only RSS from the MP, the dimension of input feature becomes N = 1, which therefore has a lower complexity order than CSE. On the other hand, CODE-NN possesses a moderate computational complexity order of $\mathcal{O}\left(N \times N_{mp} \times \prod_{i=1}^{N_L} K_i^2 \times N_{rp} \times N_{ap}\right)$, where additional complexity comes from neural network layers N_L and the corresponding neurons denoted by K_i for the *i*-th layer.



Fig. 6: Simulation results of ROMAC in SALC system in terms of (a) mean localization errors and corresponding number of clusters using different preference values of ROMAC and (b) CDF of localization error by comparing different factors in ROMAC.

The SSP-skeleton generated from ROMAC algorithm is shown in Fig. 5(b), where the black lines are skeletons and black dots are vertices. Note that the red crosses represent the RPs to be tested. In Fig. 5(c), the clustering result shows that the RPs are divided into 6 clusters by employing only the difference of RSS $d_{RSS}(i, j)$ in (9) among RPs. However, only considering the RSS difference leads to the two clusters in the bottom room overlapping each other with RPs belonging to a cluster located farther away. Fig. 5(d) shows the clustering result by only adopting the map information of each RP, i.e., using the shortest path $d_{SSP}(i, j)$ in (10), where all generated clusters will not overlap with each other due to the characteristic of SSP. Nonetheless, the clustering result is unable to reflect wireless signal propagation. Fig. 5(e) demonstrates the clustering results by utilizing the time-varying RSS difference $\delta_{i,i}$ in (11). It reveals that taking time-variation into account will only generate 2 clusters, causing them to overlap and even across the two-room partition. By implementing the proposed ROMAC scheme which considers all factors in (12), 6 clusters are automatically generated from all RPs as shown in Fig. 5(f) where the cluster heads are chosen as the locations of MPs for the corresponding clusters. The benefits of considering all three factors reveal that each clusters will not largely overlap with the others.

Fig. 6 shows the performance evaluation under different parameters of the ROMAC scheme. Fig. 6(a) demonstrates the resulting number of clusters and localization error under different values of self-similarity preference $s_{j,j}$. We evaluate proposed CODE-LR and CODE-NN in terms of localization errors with the corresponding number of clusters from RO-MAC. It can be observed that a smaller preference value will generate fewer clusters. However, a small number of clusters cannot reconstruct the database efficiently, which leads to a higher localization error in both schemes. The lowest error is reached at the preference value of $s_{j,j} = M_d(S_j) = -1.18$ in (14) under CODE-NN, which is chosen as our preference value without the limitation of the number of MP. Fig. 6(b)



Fig. 7: The CDF of localization error of proposed CODE-NN scheme in SALC system under different (a) learning rates η and (b) thresholds γ .

compares the cumulative distribution function (CDF) of localization error in the crowded environment by using different similarity measures such as difference of RSS amplitude, SSP for map information, and time-variation of RSS, and combined factors using ROMAC to choose MPs among the clustered RPs. Note that the databases in these four cases are constructed by CODE-NN, whilst the curve named original database indicates the utilization of fingerprint database in empty environments. It can be seen that the proposed ROMAC with adaptive database achieves the lowest localization error, which outperforms the other methods suffering from timevarying signal blockages and reflection. Again, this can be further emphasized with the aid of Fig. 5. As illustrated in Figs. 5(c) and 5(d), using only d_{SSP} for clustering results in separation based on geometric relationships, which neglects the other crucial factors such as d_{RSS} capturing the impact of path loss caused by indoor environments. On the other hand, $\delta_{i,i}$ takes into account dynamic signal strength fluctuations caused by the presence of people, as shown in Fig. 5(e). Disregarding these factors can lead to significant deviations in the estimated positions. To elaborate a little further, the simulations presented in this study provide a simplified scenario for evaluating the performance of the proposed clustering algorithm. It is important to acknowledge that disregarding these critical factors can have even more severe consequences in real-world experiments. Real-world environments impose additional challenges of interference and channel variations that can further affect the performance of positioning systems. Therefore, it becomes compellingly essential to consider these factors when designing and implementing positioning systems in practical scenarios.

Fig. 7(a) shows the CDF of localization error using different learning rates η in CODE-NN to reconstruct the online database. The lines from top to bottom represent $\eta = \{0.001, 0.01, 0.1, 1\}$ and original fingerprint database, respectively. The result shows that the model has the largest error when $\eta = 1$ since large learning rate potentially diverges the loss function, whilst the database adopting $\eta = 0.001$ will be



Fig. 8: Performance comparison of predicted RSS errors at different RP's locations based on (a) original databased, (b) CODE-LR scheme, and (c) CODE-NN scheme.

inefficient, since the model over-fitting is induced during data training. Moreover, using the database reconstructed under $\eta = 0.1$ can properly converge the loss function and avoid the over-fitting issue, which provides a better performance. Hence, we select $\eta = 0.1$ as the learning rate in our CODE-NN scheme in the following simulations. Fig. 7(b) shows the CDF of localization error when adopting different thresholds γ in the loss function of pre-training in (29) and fine-tuning phases to re-establish the database. The curves include parameters of $\gamma = \{0.01, 0.1, 1, 10\}$ and original database. The localization error shows that the loss function is unable to filter the outlier when γ is set to $\gamma = 10$, which causes the reconstructed database unavailable. However, the loss function will treat every output as outlier when we set the threshold as $\gamma = 0.01$, which leads to erroneous back propagation of parameters. The CDF of localization error when $\gamma = 1$ outperforms the others, which can successfully filter out the outliers. Therefore, we set the threshold as $\gamma = 1$ in our purposed CODE-NN scheme.

Fig. 8 shows the predicted RSS errors at different RPs by adopting original fingerprint database, CODE-LR, and CODE-NN. In all three subplots, the two-dimensional coordinates [x, y] is utilized to represent RP's locations described in Fig. 5(a). The predicted RSS error is calculated as $\varepsilon_{n,l}(t_o) = |\hat{\alpha}_{n,l}^{RP}(t_o) - \alpha_{GT}^{RP}(t_o)|$, where $\alpha_{GT}^{RP}(t_o)$ represents the ground truth of RSS at t_o and l = 1 is adopted by using the RSS from the AP located at the upper-left corner of Fig. 5(a). It can be observed that both CODE-NN and CODE-LR methods can reduce RSS error from the original database by reconstructing the adaptive RP database. The largest error can be seen at RP's location [x, y] = [2, 7] in Fig. 8(a), which indicates that the area with crowded people causes higher RSS errors, with the original database collecting RSS under an empty scenario. The result in Fig. 8(b) shows the regression method in CODE-LR. It can reduce most of RSS errors but has a difficulty to surpress the peak error due to linear operation of regression. Fig. 8(c) illustrates that the proposed CODE-NN can perfectly reconstruct the radio map with compellingly low RSS errors thanks to its nonlinear mapping in deep neural networks, which outperforms CODE-LR. In addition to propagation decay and cluster information in CODE-LR, CODE-NN considers timevarying effect in different environments, which achieves the lowest localization errors among the other schemes.

Fig. 9(a) shows the localization errors by adopting CsLE scheme in SALC system under different database and k values as indicated in (34). Note that the original and true databases indicate using the RSS collected from empty and crowded environments, respectively, whilst the crowded one



Fig. 9: Mean localization error of proposed CsLE scheme (a) under different k values and (b) in comparison with conventional WkNN method.

is referred as true database, since it is the realistic case to be dealt with. The result illustrates that using the original database has the largest location error under different k values; while the database reconstructed by CODE-LR and CODE-NN can significantly reduce the error. Additionally, using CODE-NN database is close to adopting the ground truth database, which means we can successfully reconstruct the radio map. It is also shown in the figure that the localization error will decrease with larger k when $k \leq 3$, whereas there is no benefit for k larger than 3 under all four cases. The reason is that a larger k means to take RPs with lower weights into consideration, which are irrelevant to the user's location. Meanwhile, smaller k may cause the chosen RPs to contain insufficient information, which leads to inaccurate predicted location. Accordingly, we select k = 3 in CsLE in the following simulations and experiments.

Fig. 9(b) shows the performance comparison between proposed CsLE and conventional WkNN schemes under k = 3with four different types of database. It can be seen that the location errors estimated based on proposed CODE-NN and CODE-LR schemes outperform that from the original database, which suffers from both time-variation and noise. On the other hand, comparably smaller error is generated from true crowded database which is mostly caused by the noise from RSS and RP distances. By taking the time-varying effect into consideration, CODE-NN effectively reconstructs



Fig. 10: Performance comparison for CDF of localization error under different schemes.

the online database and achieves similar indoor positioning performance compared to that from true database. Meanwhile, Fig. 9(b) also reveals that CsLE outperforms WkNN with the adoption of those four types of database. For estimating the user's location, WkNN may choose the RPs with similar RSS values even they are located farther away from the user location, whilst our proposed CsLE will filter those outlier RPs by adopting the cluster-based feature scaling weight in (33).

Fig. 10 illustrates the performance comparison on the CDF of localization errors among CsLE with original and true database, SALC with CODE-NN and CODE-LR, and existing schemes from [29] and [22]. Note that RWKNN adopts the original database and CSE reconstructs the database via conventional WkNN. We can observe that the RWKNN method performs the worst performance with localization error of around 2 m under CDF of 0.5 since it did not consider timevarying effects. Although the original database still encounters time-varying problem, the CsLE scheme can effectively mitigate the problems of both signal fluctuation and selecting farther RPs as neighbor nodes. Additionally, it reveals that the localization error decreases as the fingerprinting database is updated. Furthermore, our proposed SALC scheme achieves the lowest localization error of approximately 1 m under CDF of 0.5 by dynamically generating the fingerprinting database in time-varying environments.

B. Experiment Results

Experiments have been conducted to verify the effectiveness of SALC system in realistic environments. Fig. 11 shows the testing field of experiments including both the classroom and the corridor. We consider break and in-class time representing empty and crowded cases, respectively. The size of the experimental scene ² is 9.65×10.65 m², where 42 RPs are distributed

²If the new scene has a different layout, it will be necessary to redeploy the RPs to new locations. This process involves re-clustering the RPs and re-selecting the MPs. Consequently, data from different scenarios in the new scene will need to be collected to adapt ROMAC and CODE algorithms. As a future extension, transfer learning and domain adaptation approaches can be leveraged to address the challenges of retraining the system for different layouts.



Fig. 11: Testing field for experiments including both classroom and corridor at (a) break time and (b) in-class time.

with inter-RP distance of 1.2 m for database collection, and 26 TPs are determined to evaluate the positioning accuracy. We use the mobile device of ASUS Zenfone to collect RSS on each RP from the 3 APs, which are ASUS RT-AC66U operating at 2.4 GHz. We use the same model of mobile devices serving as MPs during the online phase, which means that there is no additional functional requirement and overhead for deploying MPs in our experiments. The server employs these RSS values to generate an updated radio map and to estimate the user location, which is therefore transmitted back to the mobile device. Since most of training and computing are conducted at server side, the mobile device collecting data and received positioning results has negligible computation during the process. We collect 100 samples on each RP in both break and in-class time, which takes around 1 hour in each case. Note that the crowded true database is not feasible to be collected in practical scenarios, and we establish it mainly to serve as the ground truth for performance comparison. The amount of pretraining and of training data is 4200 and 8400, respectively. The other system parameters are chosen to be the same as those in simulations.

Fig. 12(a) illustrates the layout and deployments of APs (black triangles), RPs (red crosses), and TPs (blue points), whilst Fig. 12(b) shows the result by adopting proposed ROMAC algorithm. With the consideration of hardware limitation in a practical scenario, we adjust the preference value $s_{j,j} = -5$ in (14) such that the resulting number of clusters will be limited to 3. It can be seen from Fig. 12(b) that all RPs on the corridor are in the same cluster since the SSP information in (10) is taken into account in ROMAC. The RPs in the testing classroom are divided into two clusters due to human blockage effects, which will reflect the time-varying RSS as considered in ROMAC.

Figs. 13(a) and 13(b) show the positioning results of proposed SALC system in real-time environment. The cases of using empty and crowded databases indicate that the RSS data are collected in break time and in-class time, respectively. Fig. 13(a) shows the results of noise-free environment, where the mean error of using crowded database is the largest of 2.83 m among the other databases since the RSS from the in-class time database suffers from human blocking effects. Note that the empty database with localization error of 2.16 m is treated as the ground truth database in break time. It can be observed that the errors from proposed CODE-LR and CODE-NN are respectively reduced to 2.31 and 2.33 m compared to that acquired from crowded database during inclass time. Furthermore, Fig. 13(b) illustrates the experimental



Fig. 12: (a) The experimental scene and deployments of APs, RPs, and TPs, and (b) the resulting 3 clusters and corresponding MPs by adopting proposed ROMAC algorithm.



Fig. 13: Performance comparison of proposed SALC system under different types of database during (a) break time and (b) in-class time.

results under noisy environments, where the mean localization error of using the empty database is the largest of 2.68 m since the database was not collected with signal blockage features. Note that the crowded database is treated as the ground truth one in this case resulting in 2.11 m of positioning error. It can be seen that the proposed CODE-LR method achieves the error of 2.25 m, whilst CODE-NN results in the same smallest 2.11 m error as that from ground truth crowded database.

Fig. 14 shows the overall performance comparison of proposed SALC system with existing RWKNN [29] and CSE [22] methods, where the results of mean localization error are averaged test data from both noise-free and noisy environments. The bars from left to right are RWKNN utilizing empty and crowded databases, traditional WkNN with the database constructed by CSE, proposed CsLE with empty and crowded databases, and SALC adopting CODE-LR and CODE-NN to reconstruct the databases. It can be seen that CsLE outperforms conventional RWKNN under both empty and crowded databases. Note that the localization errors are higher in crowded than empty database for both schemes. The main reason is that RSS collected for crowded data can possess high signal variations due to human movements in realistic scenarios, which incurs infeasible establishment of crowded database. Furthermore, the proposed CODE-LR and CODE-NN methods can improve localization performance by adaptively adjusting and reconstructing databases for different environments. Notice that the CSE method only constructs the database by current information with the worst performance due to the complicate multipath effects. Meanwhile, CODE-



Fig. 14: Performance comparison of proposed SALC system with existing schemes.

NN performs better than CODE-LR since it considers the nonlinear features of time-varying effects between noise-free and noisy environments.

VI. CONCLUSION

In this paper, we have designed an SALC system for indoor positioning including ROMAC, CODE, and CsLE subalgorithms. ROMAC is designed to simultaneously solve the RP clustering and MP deployment in order to monitor the time-varying problem and reconstruct the adaptive fingerprinting database. CODE establishes adaptive online database by employing linear regression and neural network techniques based on the cluster information from ROMAC scheme. At last, CsLE predicts user position by matching the user's real-time RSS with the adaptive database based on cluster information and predicted signal variation. Although we can reconstruct the radio map precisely, there still exist real-time uncertainties in experiments, such as multipath, noise, and interference from in-class and break time, which limits the performance from theory to implementation. In the future work, we will consider complicated environmental scenarios for further performance enhancement. Nevertheless, the merits of proposed SALC system can still be observed from both simulations and experiments by improving the performance of fingerprinting-based localization under practical time-varying environments.

REFERENCES

- L.-H. Shen, K.-T. Feng, and L. Hanzo, "Five facets of 6G: Research challenges and opportunities," ACM Computing Surveys, vol. 55, no. 11, 2023.
- [2] A. El-Rabbany, *Introduction to GPS: The Global Positioning System*. Artech House, 2002.
- [3] M. Pham, D. Yang, and W. Sheng, "A Sensor Fusion Approach to Indoor Human Localization Based on Environmental and Wearable Sensors," *IEEE Transactions on Automation Science and Engineering*, vol. 16, no. 1, pp. 339–350, 2019.
- [4] C. Zhou, J. Liu, M. Sheng, Y. Zheng, and J. Li, "Exploiting Fingerprint Correlation for Fingerprint-Based Indoor Localization: A Deep Learning Based Approach," *IEEE Transactions on Vehicular Technology*, pp. 5762–5774, 2021.
- [5] X. Wang, X. Wang, and S. Mao, "Deep Convolutional Neural Networks for Indoor Localization with CSI Images," *IEEE Transactions on Network Science and Engineering*, vol. 7, no. 1, pp. 316–327, 2020.
- [6] H.-C. Tsai, C.-J. Chiu, P.-H. Tseng, and K.-T. Feng, "Refined Autoencoder-Based CSI Hidden Feature Extraction for Indoor Spot Localization," in *Proc. IEEE Vehicular Technology Conference (VTC-Fall)*, 2018, pp. 1–5.

- [7] F.-Y. Chu, C.-J. Chiu, A.-H. Hsiao, K.-T. Feng, and P.-H. Tseng, "WiFi CSI-Based Device-free Multi-room Presence Detection using Conditional Recurrent Network," in *Proc. IEEE Vehicular Technology Conference (VTC-Spring)*, 2021, pp. 1–5.
- [8] X. Guo, N. R. Elikplim, N. Ansari, L. Li, and L. Wang, "Robust WiFi Localization by Fusing Derivative Fingerprints of RSS and Multiple Classifiers," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 5, pp. 3177–3186, 2020.
- [9] X. Tian, M. Wang, W. Li, B. Jiang, D. Xu, X. Wang, and J. Xu, "Improve Accuracy of Fingerprinting Localization with Temporal Correlation of the RSS," *IEEE Transactions on Mobile Computing*, vol. 17, no. 1, pp. 113–126, 2018.
- [10] X. Zhu, W. Qu, X. Zhou, L. Zhao, Z. Ning, and T. Qiu, "Intelligent Fingerprint-Based Localization Scheme Using CSI Images for Internet of Things," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 4, pp. 2378–2391, 2022.
- [11] X. Zhu, T. Qiu, W. Qu, X. Zhou, M. Atiquzzaman, and D. O. Wu, "BLS-Location: A Wireless Fingerprint Localization Algorithm Based on Broad Learning," *IEEE Transactions on Mobile Computing*, vol. 22, no. 1, pp. 115–128, 2023.
- [12] Z. Wu, X. Wu, and Y. Long, "Prediction Based Semi-Supervised Online Personalized Federated Learning for Indoor Localization," *IEEE Sensors Journal*, vol. 22, no. 11, pp. 10640–10654, 2022.
- [13] H. Liu, H. Darabi, P. Banerjee, and J. Liu, "Survey of Wireless Indoor Positioning Techniques and Systems," *IEEE Transactions on Systems, Man, and Cybernetics*, vol. 37, no. 6, pp. 1067–1080, 2007.
- [14] X. Ding, S. Sheng, J. Liu, and P. Zhou, "Efficient Probabilistic K-NN Computation in Uncertain Sensor Networks," *IEEE Transactions* on Network Science and Engineering, vol. 8, no. 3, pp. 2575–2587, 2021.
- [15] H. Wang, L. Ma, Y. Xu, and Z. Deng, "Dynamic Radio Map Construction for WLAN Indoor Location," in *Proc. IEEE Third International Conference on Intelligent Human-Machine Systems and Cybernetics* (*IHMSC*), vol. 2, 2011, pp. 162–165.
- [16] H.-X. Liu, B.-A. Chen, P.-H. Tseng, K.-T. Feng, and T.-S. Wang, "Map-aware Indoor Area Estimation with Shortest Path Based on RSS Fingerprinting," in *Proc. IEEE Vehicular Technology Conference (VTC-Spring)*, 2015, pp. 1–5.
- [17] C. Feng, W. S. A. Au, S. Valaee, and Z. Tan, "Received-Signal-Strength-Based Indoor Positioning Using Compressive Sensing," *IEEE Transactions on Mobile Computing*, vol. 11, no. 12, pp. 1983–1993, 2011.
- [18] P. Bahl and V. N. Padmanabhan, "RADAR: An In-Building RF-Based User Location and Tracking System," in *Proc. IEEE International Conference on Computer Communications (INFOCOM)*, 2000, pp. 775– 784.
- [19] M. Youssef and A. Agrawala, "The Horus WLAN Location Determination System," in Proc. ACM International conference on Mobile systems, Applications, and Services (MobiSys), 2005, pp. 205–218.
- [20] X. Zhu, W. Qu, T. Qiu, L. Zhao, M. Atiquzzaman, and D. O. Wu, "Indoor Intelligent Fingerprint-Based Localization: Principles, Approaches and Challenges," *IEEE Communications Surveys and Tutorials*, vol. 22, no. 4, pp. 2634–2657, 2020.
- [21] P. Roy and C. Chowdhury, "A Survey of Machine Learning Techniques for Indoor Localization and Navigation Systems," *Journal of Intelligent* & *Robotic Systems*, vol. 101, no. 3, pp. 1–34, 2021.
- [22] N. Hernández, M. Ocaña, J. M. Alonso, and E. Kim, "Continuous Space Estimation: Increasing WiFi-based Indoor Localization Resolution without Increasing the Site-survey Effort," *Sensors*, vol. 17, no. 1, p. 147, 2017.
- [23] M. M. Atia, A. Noureldin, and M. J. Korenberg, "Dynamic Online Calibrated Radio Maps for Indoor Positioning in Wireless Local Area Networks," *IEEE Transactions on Mobile Computing*, vol. 12, no. 9, pp. 1774–1787, 2012.
- [24] H. Zou, M. Jin, H. Jiang, L. Xie, and C. J. Spanos, "WinIPS: WiFi-Based Non-Intrusive Indoor Positioning System with Online Radio Map Construction and Adaptation," *IEEE Transactions on Wireless Communications*, vol. 16, no. 12, pp. 8118–8130, 2017.
- [25] W. Vinicchayakul and S. Promwong, "Performance Comparison between UWB and NB Propagation Models for an Indoor Localization," in *Proc. IEEE Asia-Pacific Conference on Communication (APCC)*, 2014, pp. 299–302.
- [26] Z. Liu, X. Luo, and T. He, "A Feature-Scaling-Based k-Nearest Neighbor Algorithm for Indoor Positioning Systems," *IEEE Internet of Things Journal*, vol. 3, no. 4, pp. 590–597, 2015.
- [27] C. Zhou, J. Liu, M. Sheng, Y. Zheng, and J. Li, "Exploiting Fingerprint Correlation for Fingerprint-Based Indoor Localization: A Deep Learning

Based Approach," *IEEE Transactions on Vehicular Technology*, vol. 70, no. 6, pp. 5762–5774, 2021.

- [28] W. Xue, X. Hua, Q. Li, K. Yu, W. Qiu, B. Zhou, and K. Cheng, "A New Weighted Algorithm Based on the Uneven Spatial Resolution of RSSI for Indoor Localization," *IEEE Access*, vol. 6, pp. 26588–26595, 2018.
- [29] G. Chen, X. Guo, K. Liu, X. Li, and J. Yang, "RWKNN: A Modified WKNN Algorithm Specific for the Indoor Localization Problem," *IEEE Sensors Journal*, vol. 22, no. 7, pp. 7258–7266, 2022.
- [30] L. Zhang, L. Ma, Y. Xu, and C. Li, "Linear Regression Algorithm against Device Diversity for Indoor WLAN Localization System," in *Proc. IEEE Global Communications Conference (GLOBECOM)*, 2017, pp. 1–6.
- [31] A. Ye, X. Yang, L. Xu, and Q. Li, "A Novel Adaptive Radio-Map for RSS-Based Indoor Positioning," in *Proc. IEEE International Conference* on Green Informatics (ICGI), 2017, pp. 205–210.
- [32] C. Wu, Z. Yang, and C. Xiao, "Automatic Radio Map Adaptation for Indoor Localization Using Smartphones," *IEEE Transactions on Mobile Computing*, vol. 17, no. 3, pp. 517–528, 2017.
- [33] J. Yin, Q. Yang, and L. M. Ni, "Learning Adaptive Temporal Radio Maps for Signal-Strength-Based Location Estimation," *IEEE Transactions on Mobile Computing*, vol. 7, no. 7, pp. 869–883, 2008.
- [34] D. Zhu, H. Zhang, and W. Feng, "Research on the Construction of Radio-Map Based on Support Vector Regression," in *Proc. IEEE International Conference on Instrumentation and Measurement, Computer, Communication and Control*, 2014, pp. 77–80.
- [35] H. Zou, Y. Zhou, H. Jiang, B. Huang, L. Xie, and C. Spanos, "Adaptive Localization in Dynamic Indoor Environments by Transfer Kernel Learning," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.
- [36] L.-H. Shen, K.-J. Chen, A.-H. Hsiao, and K.-T. Feng, "BTS: Bifold Teacher-Student in Semi-Supervised Learning for Indoor Two-Room Presence Detection Under Time-Varying CSI," 2023. [Online]. Available: https://arxiv.org/abs/2212.10802
- [37] P. Barsocchi, S. Lenzi, S. Chessa, and G. Giunta, "A Novel Approach to Indoor RSSI Localization by Automatic Calibration of the Wireless Propagation Model," in *Proc. IEEE Vehicular Technology Conference* (VTC), 2009, pp. 1–5.
- [38] D. Berrar, "Bayes' Theorem and Naive Bayes Classifier," *Encyclopedia of Bioinformatics and Computational Biology*, pp. 403–412, 2018.
- [39] E. Gokcay and J. C. Principe, "Information Theoretic Clustering," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 24, no. 2, pp. 158–171, 2002.
- [40] C.-J. Chiu, K.-T. Feng, and P.-H. Tseng, "Spatial Skeleton-Enhanced Location Tracking for Indoor Localization," in *Proc. IEEE Wireless Communications and Networking Conference (WCNC)*, 2017, pp. 1–6.
- [41] H. Choset and J. Burdick, "Sensor-Based Exploration: The Hierarchical Generalized Voronoi Graph," *The International Journal of Robotics Research*, vol. 19, no. 2, pp. 96–125, 2000.
- [42] M. Awad and R. Khanna, "Support Vector Regression," *Efficient Learning Machines*, pp. 67–80, 2015.
- [43] A. B. Owen, "A Robust Hybrid of Lasso and Ridge Regression," Contemporary Mathematics, vol. 443, no. 7, pp. 59–72, 2007.