IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received 27 February 2025, accepted 9 April 2025, date of publication 15 April 2025, date of current version 25 April 2025.

RESEARCH ARTICLE

CIELO: Class-Incremental Continual Learning for Overcoming Catastrophic Forgetting With Smartphone-Based Indoor Localization

AKHIL SINGAMPALLI, (Graduate Student Member, IEEE),
DANISH GUFRAN, (Member, IEEE), AND **SUDEEP PASRICHA**, (Fellow, IEEE)
Department of Electrical and Computer Engineering, Colorado State University, Fort Collins, CO 80523, USA

Corresponding author: Akhil Singampalli (akhil.singampalli@colostate.edu)

This work was supported in part by the National Science Foundation under Grant CNS-2132385.

ABSTRACT With dynamically evolving indoor environments, class-incremental learning (CIL) plays a crucial role in enabling indoor localization systems to adapt to new indoor areas. However, CIL poses additional challenges such as catastrophic forgetting, where patterns from previously learned paths are overwritten by data from new paths, and high data storage demands on the edge server, which must retain extensive localization data, resulting in high memory and power consumption overheads. To address these challenges, an effective solution must support CIL with indoor paths while mitigating catastrophic forgetting and reducing storage overheads on the edge server. To the best of our knowledge, CIELO is the first framework to address these challenges in the domain of indoor localization. It introduces a novel CIL approach that integrates an innovative representation memory management (RMM) policy with crowdsourcing to enable high-accuracy localization while significantly reducing catastrophic forgetting and data storage requirements. Through extensive experimental evaluations conducted across multiple real-world paths and devices, our results demonstrate that CIELO improves indoor localization accuracy by up to 29.4× with up to 60 newly introduced classes (locations) across paths, reduces data storage by up to 1.75×, and power consumption by up to 1.69× on the edge server, compared to state-of-the-art solutions.

INDEX TERMS Continual learning, class-incremental learning, catastrophic forgetting, indoor localization, Wi-Fi fingerprinting.

I. INTRODUCTION

Indoor localization systems are being adopted across a wide range of applications, including augmented and virtual reality (AR/VR), smart home automation, asset tracking, and indoor navigation [1]. This widespread adoption is fueling the rapid growth of the indoor localization market, which is projected to reach USD 83.1 billion by 2030, as demand rises for precise positioning solutions across diverse sectors [2].

To aid with indoor localization across environments where global positioning system (GPS) signals are absent (e.g., buildings, urban areas, underground mines), industries are exploring a range of wireless radio frequency (RF) technologies (e.g., Wi-Fi, Bluetooth, and Ultra-Wideband (UWB)) [3], [4], [5], channel characteristics (e.g., channel state information (CSI), received signal strength (RSS) [6]) and other localization techniques (e.g., trilateration, dead reckoning, fingerprinting [7]). Out of these, Wi-Fi RSS-based fingerprinting has emerged as a prominent choice for smartphone-based indoor localization for users [8]. This can be largely attributed to the widespread availability of Wi-Fi in indoor locales and its seamless integration with modern mobile devices [9]. RSS fingerprinting involves capturing signal strength values from multiple Wi-Fi access points (APs) to create unique "fingerprints" associated with specific locations or reference points (RPs) within an indoor environment. These fingerprints are then trained using machine learning (ML) models to predict a device's location

The associate editor coordinating the review of this manuscript and approving it for publication was Liang-Bi Chen.

by learning patterns in the RSS data, making the approach suitable for real-time deployment in mobile devices.

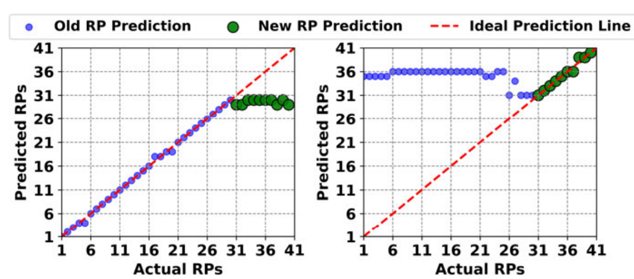
Wi-Fi RSS fingerprinting-based indoor localization systems typically operate in two phases: offline and online [7]. During the offline phase, RSS fingerprints are collected at various RPs, and an ML model is trained to learn location-specific patterns. In the online phase, the trained ML model is deployed on mobile devices, where it predicts the device's location in real-time using RSS values collected at unknown RPs. Despite the theoretical simplicity and effectiveness of this approach, indoor environments are inherently dynamic, which introduces noise into fingerprints, i.e., RSS fingerprint measurements are prone to variations caused by factors such as obstacles, human movement, and environmental phenomena such as multipath fading effects. Such variations degrade the performance of static ML models, which struggle to adapt to continuous changes in the environment, limiting their accuracy in dynamic scenarios.

Moreover, extending a deployed ML model to localize new paths (those not included in offline training) or to adapt to layout changes presents several challenges. The model relies on pre-stored RSS fingerprints for training, and incorporating new path data requires: 1) expanding this fingerprint database to accommodate new RP data, which increases storage requirements, and 2) retraining the ML model which adds computational and time overheads. Adapting to the vagaries of dynamic environments requires continuous updates to the fingerprint database and ML model, a process that remains an open challenge for robust and efficient indoor localization.

Continual learning (CL) represents a promising solution for this problem, offering the capability to update an ML model dynamically with new data while retaining previously learned knowledge [10]. Unlike conventional ML models, which are designed to work with static data distributions, CL focuses on adapting to dynamic and evolving data distributions. By integrating crowdsourcing into this process, new indoor localization data can be gathered from mobile users over time. This data contributes to updating a centralized ML model, referred to as the global model (GM), which enables the system to continuously learn and adapt to new RPs or locations not covered during the initial offline training phase, significantly enhancing its applicability to evolving paths and new environments.

CL scenarios are often categorized based on how the input and output distributions evolve. For instance, some approaches adapt to changes in input data distributions without expanding the outputs (not introducing new RPs), while others require the system to handle entirely new outputs, such as additional RPs [11]. Among these, Class-Incremental Learning (CIL) is considered the most challenging [12], as it requires the GM to predict an ever-growing set of outputs, such as new RPs, without explicitly knowing which part of the data belongs to previous or new distributions. Implementing CIL for indoor localization introduces two major challenges: catastrophic forgetting and data storage overheads. Catastrophic forgetting arises when continual updates to the GM result in the loss of knowledge from previously learned RPs [13], [14], [17]. As new RPs are introduced, the GM may become biased towards these new locations, effectively "forgetting" earlier learned RPs and causing a decline in localization accuracy. A naive solution would involve storing data for all previously learned RPs and periodically retraining the GM on all of this data. However, this approach is impractical due to the high data storage and computational costs associated with it.

To illustrate the impact of catastrophic forgetting, we conducted an experiment using a four-layer deep neural network (DNN) as the GM for Wi-Fi RSS fingerprinting-based indoor localization in a real building. Initially, the DNN was trained offline on RSS fingerprints from 30 RPs (0 to 30). Subsequently, it was updated with data from 10 additional RPs (31 to 40) crowdsourced from users' mobile devices. Fig. 1 presents the results of this experiment. In Fig. 1(a), the DNN is trained exclusively on the initial 30 RPs and it performed well when tested on these same RPs, demonstrating accurate classification. However, the DNN failed to accurately classify the new RPs, misinterpreting them as existing classes, as the new RPs were not a part of the initial training data. In Fig. 1(b), the DNN was retrained exclusively with data from the 10 new RPs, resulting in improved classification accuracy only for the new RPs but a marked decline in performance for the original 30 RPs. This outcome suggests that the DNN replaced its prior knowledge of the initial 30 RPs with information about the newly added 10 RPs, a clear demonstration of catastrophic forgetting. These results highlight the pressing need for a method that can effectively integrate new RPs while preserving knowledge of previously learned RPs, without retraining the GM on the entire dataset repeatedly.



(a) (b)

FIGURE 1. **Illustration of catastrophic forgetting using a four-layer DNN. (a) DNN predictions after training on initial 30 RPs (0 to 30). (b) DNN predictions after retraining exclusively on 10 new RPs (31 to 40).**

To address these important challenges, we propose CIELO (Class-Incremental Learning for Indoor Localization), which to the best of our knowledge, is the first framework to introduce CIL for indoor localization. CIELO is designed to address the unique challenges of this domain, incorporating an innovative CIL approach to continually update the GM with new RPs while retaining prior knowledge, with low data storage overheads. To achieve efficient on-device inference, CIELO leverages an edge server for periodic model updates and storage management, ensuring real-time localization. Our key contributions in this work are:

- We propose a novel CIL framework that incrementally learns new RPs while preserving accuracy on previously learned RPs, enabling the GM to learn new indoor paths.
- We present a novel representation memory management (RMM) policy to mitigate catastrophic forgetting while capping data storage requirements.
- We optimized CIELO for resource-limited mobile devices and low edge server overheads.
- We conduct comprehensive real-world evaluations using data collected from multiple mobile devices and indoor environments, benchmarking CIELO against state-of-the-art methods.

II. RELATED WORK

Wi-Fi fingerprinting-based indoor localization has gained significant attention, with research conferences like IEEE IPIN [19] and competitions hosted by companies such as Microsoft [21], driving innovation and advancing the field. Traditional ML methods such as KNN [22], HMM [23], and GPC [24] effectively address RSS fluctuations caused by factors such as human interference and signal shadowing [25]. Building on these, deep learning frameworks like DNNLOC [26], CNNLOC [27], and ANVIL [28] leverage neural networks to improve feature extraction and localization accuracy. However, these models are inherently static and lack the ability to dynamically adapt to environmental changes. These advancements demonstrate progress but highlight the need for more adaptive solutions in dynamic environments.

Federated learning (FL) has emerged as an effective solution to address the limitations of traditional indoor localization solutions, by learning to aggregate and adapt to noisy information contributed by users. Early FL frameworks like KRUM [29] employed Multi-Layer Perceptrons (MLPs) and Euclidean distance-based filtering to enable localization. FEDLOC [30] improved upon KRUM by adopting a DNN as the GM and using the popular FedAvg (a technique to update the GM), enhancing resilience to RSS fluctuations but also introducing biases from noisy local data updates from users. FEDHIL [15] further refined this approach with domain-specific aggregation to reduce the impact of noisy updates, improving robustness. However, FEDHIL, like other FL frameworks, remains limited in its ability to handle new RPs or classes incrementally. To address this, CL techniques have been explored across various domains to enhance FL by enabling models to learn sequentially. However, a key challenge of CL is its reliance on full retraining, which our work aims to overcome in the context of CIL.

Among the different CL paradigms, CIL is particularly relevant for indoor localization, as it requires a GM to integrate newly mapped RPs into an expanding localization space. This setting is widely studied in domains like object recognition and language modeling, where it is critical for models to classify new categories without prior knowledge of task boundaries [13], [14]. A major challenge in CIL is catastrophic forgetting. To address this, several approaches have been explored in domains outside of indoor localization. Elastic Weight Consolidation (EWC) [16] constrains updates to GM weights critical for earlier tasks, helping mitigate forgetting. However, EWC requires computationally intensive weight importance matrices, making it impractical for resource-limited devices. Progressive Neural Networks (PNNs) [31] create new branches for each task, isolating knowledge and preventing interference, but it results in significant memory and computational demands. Learning Without Forgetting (LwF) [18] preserves outputs from earlier tasks to enable incremental learning, but its reliance on consistent task-specific outputs often fails in dynamic and non-IID (non-independent and identically distributed) environments, conditions commonly encountered in Wi-Fi RSS fingerprinting. Other methods, such as Generative Replay (GR) [32], generate synthetic data to simulate previous tasks, reducing the need for storing large datasets. However, GR's reliance on computationally intensive generative models makes it challenging for resource-limited devices. Incremental Classifier and Representation Learning (iCaRL) [20] uses a similarity-based sample selection approach to retain a small subset of (representative) data, offering memory efficiency. But iCaRL struggles to handle evolving data distributions in dynamic environments.

In summary, the methods EWC [16], PNNs [31], LwF [18], GR [32], and iCaRL [20] represent diverse approaches that attempt to mitigate catastrophic forgetting to some extent; however, it remains an open challenge, particularly in dynamic environments where new classes are introduced incrementally. Moreover, these approaches often rely on significant data storage or computationally intensive resources, limiting their applicability to indoor localization systems. To address these challenges, we present CIELO, the first Class-Incremental Learning (CIL) framework designed specifically for indoor localization. CIELO integrates CL principles with resource-efficient strategies, enabling it to handle evolving environments and new RPs while maintaining robust performance.

III. CONTINUAL LEARNING: OVERVIEW

CL enables the GM to adapt to new data, overcoming the limitations of conventional training methods that assume simultaneous access to the entire dataset [10], [33]. This makes it relevant in dynamic environments where data evolves over time. CL scenarios are typically categorized into three types [12] based on how input ($\boldsymbol{x}$) and output ($\boldsymbol{y}$) distributions evolve over time:

1) Domain-Incremental Learning (DIL):

Here for $\boldsymbol{f}(\boldsymbol{x}) = \boldsymbol{y}$, $\boldsymbol{x}$ evolves but $\boldsymbol{y}$ remains fixed [34], [35]. The GM adapts to changes in input distributions such as RSS fluctuations, but as $\boldsymbol{y}$ remains fixed, it does not support learning new classes.

2) Task-Incremental Learning (TIL):

Here, $f_t(x) = y_t$ where both x and y evolve, and the task identity (t) is provided. This knowledge enables the GM to apply task-specific adaptations, such as using separate classifiers or specialized modules for each task [36]. However, the reliance on t limits its applicability in scenarios where such information is unavailable.

3) Class-Incremental Learning (CIL):

Here for $f(x) = \{y_1, y_1, \ldots y_n\}$ both x and y evolve, but task identity t is not provided. The GM must handle an expanding output space (e.g., any number of new RPs) while retaining its ability to map x to y for previously learned RPs. This is the most challenging type of CL problem, as the GM must balance integrating new knowledge while preserving older one, for arbitrarily large numbers of new RPs [37].

A. CIL IN INDOOR LOCALIZATION

In CIL for indoor localization, a GM designed for the problem must learn new classes incrementally and expand its capabilities as new classes become available. Unlike traditional ML models, which are trained on a fixed number of classes, CIL systems dynamically update the GM as new classes are introduced by users. In the context of Wi-Fi RSS fingerprinting-based indoor localization, adding a new class corresponds to introducing a new RP to the GM. Each RP is treated as a unique "class" in the classification task, and CIL allows the GM to incorporate these new RPs during the online phase.

During the offline phase, the GM is initially trained with RSS fingerprints collected from a predefined set of RPs (crowdsourced from users initially), denoted as C_{old} on an edge server. This GM is distributed to the user's mobile devices to enable them to perform indoor localization. When new RPs (C_{new}) are introduced in the online phase (e.g., after crowdsourcing them from users traversing new locations), the GM's architecture must be updated to include the additional classes. This requires modifying the output layer of the GM to accommodate the expanded set of classes. The GM's output layer typically uses a softmax function, which converts RSS values (logits) into probabilities for each class. The logits (Z) for the C_{old} are calculated as shown in (1):

$$Z_{old} = W_{old} * H + B_{old} \tag{1}$$

where Z_{old} is a vector containing the logits for C_{old}, W_{old} is the weight matrix of the output (classification) layer, H is the feature vector produced by the final hidden layer of the GM summarizing the RSS fingerprint input, and B_{old} is the bias vector corresponding to the C_{old} classes. The softmax function is then applied to convert Z_{old} into probabilities (Y_i) for class i, as shown in (2):

$$Y_i = \frac{exp(Z_i)}{\sum_{j=1}^{C_{old}} exp(Z_i)}, \ For\ i = \{1, 2 \ldots\ C_{old}\} \tag{2}$$

When C_{new} is introduced, the GM's output layer must be expanded to include the additional classes. This involves updating the W_{old} and B_{old} of the output layer to accommodate C_{new}, as shown in (3):

$$W_{old,\ new} = [W_{old},\ \ W_{new}];\ B_{old,\ new} = [B_{old},\ \ B_{new}] \tag{3}$$

where W_{new} and B_{new} are the weight matrix and bias vector of C_{new}. $W_{old,\ new}$ is this updated output layer and the new logits (Z_{new}) are computed as shown in (4):

$$Z_{new} = W_{old,new} * H + B_{old,new} \tag{4}$$

where Z_{new} now contains logits for both the previously learned classes and the newly added ones. This process enables the gm to classify rss fingerprints from both the original rps and the newly introduced rps, thereby dynamically expanding its localization classes. However, simply expanding the gm's output layer in this manner introduces challenges, such as catastrophic forgetting.

B. CATASTROPHIC FORGETTING

A significant challenge in CIL, especially in dynamic environments like indoor localization, is catastrophic forgetting. This challenge arises when new RPs are introduced, requiring the GM to update its representation layers (such as output layer) to capture the unique signal patterns of these new RPs. However, without access to historical data, these updates often overwrite previously learned representations. This leads to a shift in the GM's weights, favoring the new RPs at the expense of earlier ones. As a result, the GM exhibits bias toward newly added classes (new RPs), while "forgetting" previously learned classes. This bias compromises localization accuracy for prior locations, as shown earlier in Fig. 1. The gradual loss of earlier knowledge as new classes are introduced further highlights the limitations of conventional CIL methods [38]. This challenge underscores the fundamental trade-off between stability and plasticity in continual learning. Plasticity enables the model to adapt to new RPs, but excessive plasticity causes forgetting of past knowledge. Conversely, stability helps retain earlier knowledge, but excessive stability may hinder the integration of new RPs. Striking an optimal balance between these two is crucial for ensuring consistent and accurate localization performance across all RPs, both old and new. To overcome this limitation and facilitate seamless learning of new classes, we propose a new CIL method, which is described next.

IV. CIELO FRAMEWORK

The CIELO framework begins with the offline phase, as illustrated in Fig. 2. During this phase, RSS fingerprints are labeled for various RPs. The collected data is stored in the edge server as a representation memory (RM), which initially contains data for C_{old} (the original RPs). This C_{old} dataset is also used to train the GM during the offline phase. Once the GM is trained on C_{old}, the trained GM is deployed on user mobile devices (smartphones), where it is used to infer a user's location during the online phase. In the online phase, when a user encounters a new RP during the online phase, the system transitions into the CIL phase. At this

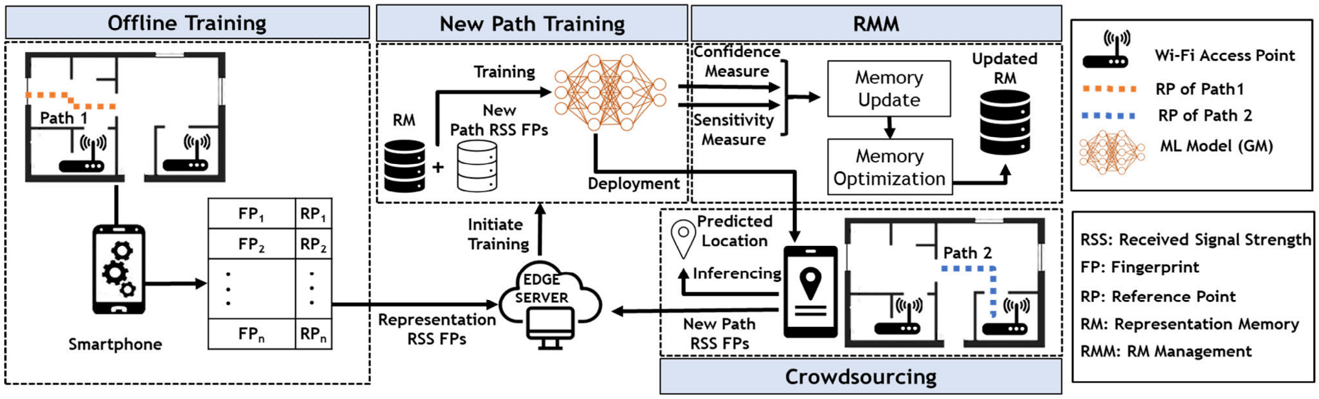


FIGURE 2. Overview of the CIELO framework for class incremental learning in Wi-Fi RSS fingerprinting-based indoor localization.

stage, the user uploads the RSS fingerprints collected at the new RP (C_{new}) to the edge server. The edge server processes this new data by combining it with samples from the existing RM (C_{old}) to create a balanced training dataset that includes both old and new RPs ($C_{old} + C_{new}$). The GM is then retrained on this updated dataset to incorporate the new RPs. However, directly merging $C_{old} + C_{new}$ introduces several challenges: 1) It increases the risk of catastrophic forgetting, where the GM loses knowledge of previously learned RPs as it learns the new ones; 2) As more new RPs are added, the RM becomes bulky and large, leading to higher storage requirements and increased GM retraining overheads. These challenges necessitate a strategic memory management policy to effectively manage the RM.

CIELO addresses these challenges with a novel representation memory management (RMM) policy, which operates via two key processes: Memory Update and Memory Optimization. These processes ensure that the RM retains only the most diverse and representative data samples, preserving critical features from previous RPs while minimizing storage and computational overheads. By doing so, CIELO enables CIL with minimal data storage (RMM footprint) while effectively mitigating the catastrophic forgetting challenge, making it highly suitable for dynamic indoor localization systems.

A. REPRESENTATION MEMORY MANAGEMENT (RMM)

The RMM policy is a critical component of the CIELO framework that ensures effective handling of data while mitigating the challenges of catastrophic forgetting and storage overhead. As shown in Fig. 3, RMM operates by evaluating and managing the RM through two main processes: memory update and memory optimization. These processes work in tandem to retain only the most diverse and representative data samples, enabling the GM to integrate new RPs seamlessly while preserving knowledge of previously learned ones. The RMM evaluates RSS fingerprints using two measures—Confidence Measure and Sensitivity Measure—to determine the importance of each fingerprint. This evaluation helps prioritize data samples for retention in the RM, ensuring that the memory remains compact while maintaining its relevance to both previously learned and newly added RPs.

1) MEMORY UPDATE

The memory update process is responsible for selecting new RSS fingerprint samples for inclusion in the RM. This selection is based on the importance of the samples, as determined by the confidence measure and sensitivity measure metrics, which ensure an optimal balance between plasticity and stability and operate as follows:

a: CONFIDENCE MEASURE

The confidence measure evaluates how well the GM can classify a given RSS fingerprint based on its class activations. For each sample (x), the GM generates class activations (shown in green in Fig. 3), which represent the probabilities assigned to each class. The key metric used in this measure is the confidence score (CS), defined as the difference between the highest and second-highest class activations. Samples with low CS are prioritized for inclusion in the RM, as they indicate uncertainty in classification and are likely to be critical for improving the GM's performance. The CS is computed using (5):

$$CS(x) = p_1(x) - p_2(x) \tag{5}$$

where $p_1(x)$ and $p_2(x)$ are the top two predicted probabilities for RSS fingerprint sample x. By retaining the least confident samples, the RM ensures that CIELO focuses on refining its learning for uncertain RPs, improving its ability to classify new RPs correctly.

b: SENSITIVITY MEASURE

The sensitivity measure assesses the impact of each sample (x) on the GM's parameters (such as weights) by introducing small perturbations to the GM. This perturbation is calculated using the Frobenius norm of the weight difference between the baseline GM weights—before training on new RPs and the updated GM weights—after training on new RPs. The Frobenius norm determines the magnitude of the perturbation, ensuring that larger changes in the GM's

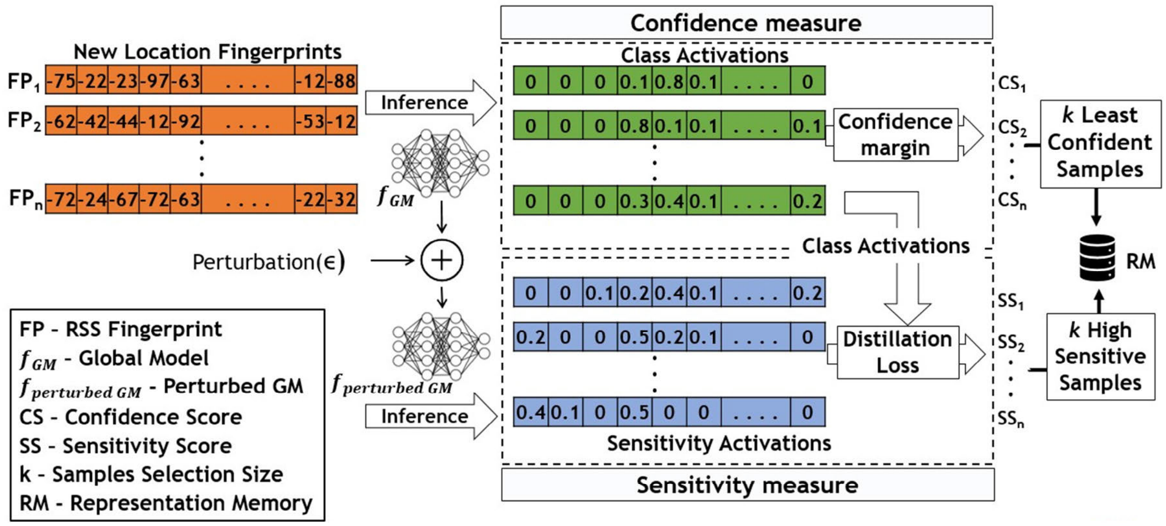


FIGURE 3. Confidence and sensitivity measure for memory management.

weights result in proportionally larger perturbations, as described in (6):

$$W_{perturbed} = W + \|W - W^b\|_F * N\left(0, 1\right) \tag{6}$$

The perturbed GM is then evaluated to observe changes in its class activations. Samples that cause significant changes in the activations are identified as highly sensitive, indicating their importance for maintaining the GM's stability and generalization capabilities. In (6), W represents the updated GM weights after training on new RP, W^b represents the baseline GM weights before training on new RP, $\|W - W^b\|_F$ is the Frobenius norm of the weight difference, and $N\left(0, 1\right)$ is noise from a standard normal distribution, which introduces variability to simulate diverse perturbations, ensuring robustness and preventing deterministic bias in sensitivity analysis. The Frobenius norm $\|W - W^b\|_F$ is computed as:

$$\|W - W^b\|_F = \sqrt{\sum_i \sum_j \left(W^{ij} - W^{b,ij}\right)^2} \tag{7}$$

The sensitivity of a sample (x) is evaluated using the distillation loss, measured as the Kullback-Leibler (KL) divergence between the GM's predictions before and after applying the perturbation, as described in (8):

$$SS(x) = KL(f_{GM}(x) \| f_{perturbed\ GM}(x)) \tag{8}$$

The KL divergence measures how much one probability distribution diverges from another. In our problem context, it reflects how sensitive the GM's predictions are to changes in the weights caused by the perturbation. A higher KL divergence indicates that the sample significantly influences the GM's predictions, marking it as highly sensitive. Samples with the highest sensitivity $SS(x)$ are prioritized for inclusion in the RM because they play a crucial role in preserving the GM's stability across both old and new data. High-sensitivity samples prevent the GM from converging to local minima, which optimize for learning new data at the cost of forgetting patterns learned from old data. Instead, they act as stabilizers, anchoring the GM closer to the global minima, where performance is balanced across both old and RPs. This ensures robust learning, preventing catastrophic forgetting while integrating new knowledge effectively.

By leveraging confidence and sensitivity scores, CIELO ensures that the RM is populated with the most critical and informative samples. Selecting the k least confident samples ensures that uncertain samples, which can benefit most from retention, are prioritized, enhancing plasticity by refining the model's adaptation to new RPs and addressing data storage challenges by reducing the need to store all data. At the same time, selecting the k most sensitive samples helps mitigate catastrophic forgetting by retaining key samples that stabilize the GM and maintain its ability to generalize across both old and new RPs. This dual k sample selection strategy optimizes memory usage while ensuring robust CIL for evolving indoor localization tasks.

2) MEMORY OPTIMIZATION

The memory optimization process addresses the challenge with the RM growing excessively large as more RPs are added. This process ensures that the memory remains compact and efficient by eliminating redundant or less informative samples while retaining diversity and representativeness in the data. CIELO defines a memory budget (M_b), which specifies the maximum size of the RM. This budget is critical to ensure efficient memory utilization, particularly in resource-limited environments. It prevents excessive memory usage while maintaining enough impactful samples to support the GM's generalization across both old and new RPs. When the memory buffer reaches M_b, CIELO prioritizes the removal of high-confidence, low-sensitivity samples per RP, as these are well-learned and contribute less to further learning. Retaining too many high-confidence samples can also lead to overfitting, where the GM becomes overly specialized to specific instances and struggles to generalize to new or unseen data. However, to avoid catastrophic forgetting, CIELO guarantees that at least one sample per RP is retained, even if it is a high-confidence sample. This safeguard ensures that no RP is completely removed, preserving the GM's ability to generalize across all RPs. By periodically pruning less critical samples while adhering to M_b, CIELO balances memory efficiency and generalization, supporting robust CIL without exceeding storage constraints.

V. EXPERIMENTAL RESULTS

A. EXPERIMENTAL SETUP

To evaluate the proposed CIELO framework, experiments were conducted in two buildings with distinct layouts: Building 1, containing 60 RPs and up to 193 visible APs, and Building 2, containing 90 RPs and up to 78 visible APs. Each RP was spaced at a 1-meter interval to ensure adequate granularity for precise indoor localization. RSS values were standardized to a range between 0 dBm (strongest signal) and -100 dBm (weakest signal), maintaining consistency across varying signal strengths.

Testing was performed using two mobile devices carried by users: HTC U11 and Samsung Galaxy S7. Self-device testing involved training and testing using data collected from the same device, while cross-device testing assessed the GM generalizability by training on data from one device and testing on the data from another device. During the offline phase, the initial GM was trained using data from 30% of the RPs in each building. The remaining 70% of RPs were incrementally introduced during the CIL phase. At each RP, six RSS fingerprints were collected for training and one RSS fingerprint was reserved for testing.

The performance of CIELO was assessed using the Euclidean distance (ED) method as defined in (9):

$$ED = \sqrt{(X_{pred} - X_{true})^2 + (Y_{pred} - Y_{true})^2 + (Z_{pred} - Z_{true})^2} \tag{9}$$

ED is used to measure the distance between predicted location (X_{pred}, Y_{pred}, Z_{pred}) and ground-truth location (X_{true}, Y_{true}, Z_{true}), offering a direct and interpretable metric for evaluating localization accuracy, where smaller distance values indicate better performance. It serves as a reliable measure to assess and compare CIELO's performance across different scenarios.

The GM used in CIELO is a Multi-layer Perceptron (MLP) with ReLU activations in each layer. We explored CIL for indoor localization across two separate scenarios in two different buildings. For Building 1, the GM consists of two hidden layers: the first hidden layer has 144 neurons, and the second hidden layer has 67 neurons. For Building 2, the first hidden layer has 58 neurons, and the second hidden layer has 27 neurons. The number of output layer neurons varies as a function of the changing number of RPs to be predicted over time. The GM resulted in a maximum of 77,541 total parameters for Building 1 and a maximum of 14,801 parameters for Building 2. These MLP architectures were empirically designed to balance complexity and deployment feasibility, incorporating insights from state-of-the-art indoor localization methods. Offline training employed the Adam optimizer with a 0.01 learning rate using sparse categorical cross-entropy loss, while incremental retraining employed a learning rate of 0.1.

B. SENSITIVITY ANALYSIS: DETERMINING SAMPLE SELECTION SIZE (k)

In our CIL framework, selecting the number of samples per class (k) to preserve in the RM is crucial for balancing memory usage and localization performance. To determine the optimal k, we experimented with saving 0 to 6 samples per class in the RM and analyzed localization accuracy. As shown in Fig. 4, increasing k reduces localization error by providing the GM with a broader representation of signal variations, enhancing generalization. However, improvements diminish beyond $k = 4$. Retaining 5 or 6 samples slightly improves accuracy but requires higher memory overheads. Choosing $k = 4$ strikes a balance between memory usage and performance, offering significant accuracy gains while keeping memory manageable. Beyond $k = 4$, additional samples yield marginal accuracy improvements that do not justify the increased resource usage. For instance, Building 1 benefits slightly from $k = 5$ or $k = 6$, but the trade-off makes $k = 4$ more practical.

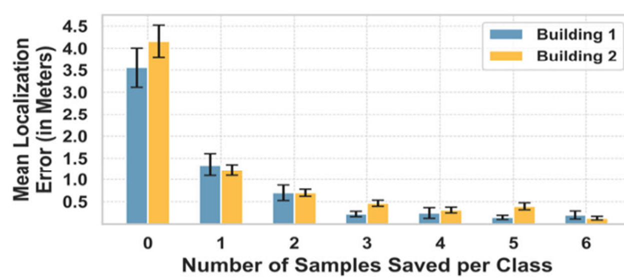


FIGURE 4. Mean localization error with increasing number of samples per class.

C. SENSITIVITY ANALYSIS: DETERMINING MEMORY BUDGET (M_b)

The memory buffer size (M_b) directly impacts the trade-off between memory storage in RM and GM performance in CIL. To evaluate its impact, we conducted experiments with M_b allocations set to 0%, 25%, 50%, 75%, and 100% of the total memory required to store all samples for a given floor plan, analyzing their effect on mean localization error across the two buildings and testing devices.

As shown in Fig. 5, increasing M_b improves accuracy, with the highest error at 0% due to the absence of representative data for earlier RPs. In Building 1, mean error drops significantly at 25% allocation, with diminishing returns beyond this point. This reflects CIELO's memory optimization strategy, where retaining only the most impactful and diverse samples enhances performance. In Building 2, a more gradual improvement in error reduction suggests that the environment may require a larger memory allocation to adequately represent the signal diversity. However, increasing M_b beyond a certain point risks redundancy and inefficiency. At 100% allocation for instance, the error is slightly higher compared to 25%-75% due to overfitting caused by retaining redundant high-confidence, low-sensitivity samples that do not contribute meaningfully to generalization. These redundant samples can cause the GM to become overly specialized, reducing its ability to generalize to new or unseen data. This issue is more evident in Building 2, which undergoes more retraining cycles due to its larger number of RPs. Allocating 25% of memory, paired with 4 samples per class, strikes a balance by preventing overfitting, optimizing RM use with 75% reduction in data storage, and maintaining high localization accuracy.

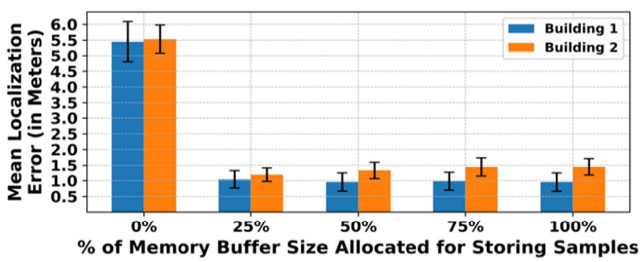


FIGURE 5. Mean localization error with increasing M_b.

D. COMPARISON WITH STATE-OF-THE-ART

To evaluate CIELO's performance against state-of-the-art CIL and indoor localization frameworks, we conducted a comparative study involving Elastic Weight Consolidation (EWC) [16], Incremental Classifier and Representation Learning (iCaRL) [20], and Learning Without Forgetting (LwF) [18], which represent diverse state-of-the-art CIL strategies from prior work. For fair comparison, all baselines were initialized with the same GM, ensuring identical starting knowledge and eliminating biases from initialization, model capacity, or pre-training. Thus, performance differences stem solely from the CIL strategies used. We also compared FEDHIL [15], which supported distributed and crowdsourced learning for indoor localization. The results of this study are shown in Fig. 6 (for Building 1) and Fig. 7 (for Building 2). The experiments were performed across the two buildings on paths with different layouts, using Samsung Galaxy S7 and HTC U11 devices to assess both self and cross-device testing. Mean localization error was measured as a function of the cumulative number of RPs learned by the GM to determine forgetting rate, with new RPs introduced in increments of 5. Across all configurations in Building 1 and Building 2, for self-device and cross-device testing, CIELO consistently outperformed other frameworks, achieving lower mean localization errors as more RPs were learned. CIELO's RMM policy allows dynamic sample selection based on confidence and sensitivity, enabling it to integrate new RPs effectively while retaining accuracy, demonstrating strong resilience to catastrophic forgetting.

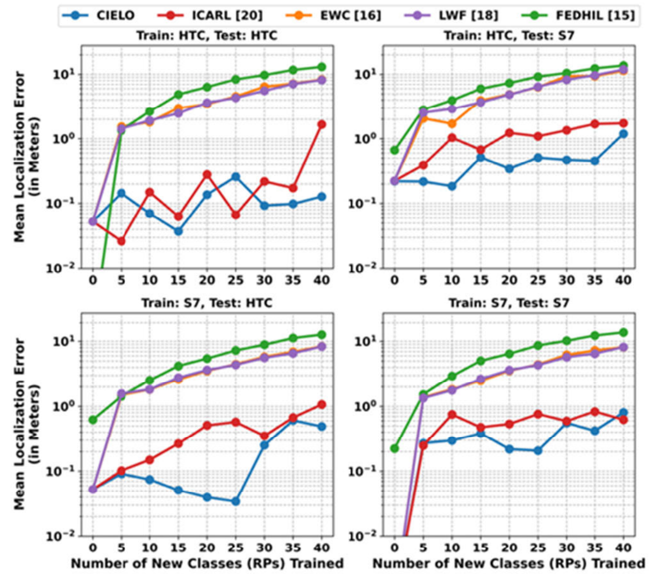


FIGURE 6. Comparison of mean localization errors across different state-of-the-art frameworks under both self and cross-device testing for Building 1.

In Building 1, as shown in Fig. 6, with self-device testing (S7), CIELO maintained significantly lower localization errors compared to other methods, which showed a larger increase in error as new RPs were learned. In cross-device testing (training on S7, testing on HTC), CIELO exhibited strong generalization, maintaining consistent accuracy on a different device. In contrast, EWC [16], iCaRL [20], and LwF [18] struggled with higher errors, indicating challenges with non-IID data and incremental adaptation.

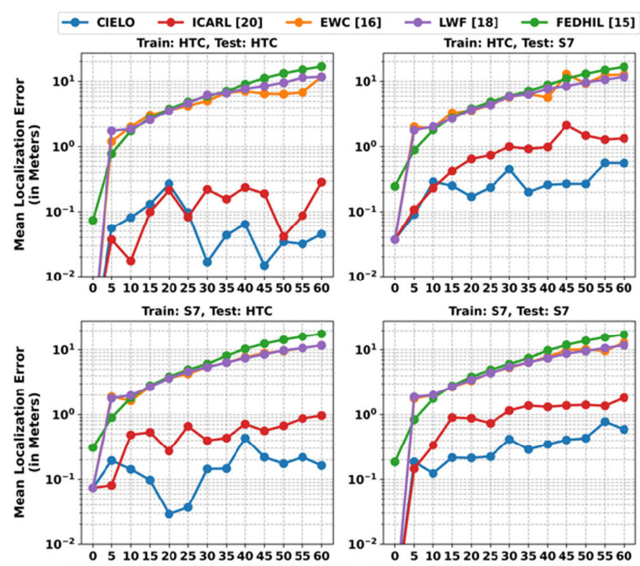


FIGURE 7. Comparison of mean localization error across different state-of-the-art frameworks under both self and cross-device testing for Building 2.

In Building 2, as shown in Fig. 7, with its higher number of RPs, CIELO's advantage persisted, as it maintained a steady, low-error trajectory while other frameworks displayed progressively higher mean localization errors. These results highlight CIELO's robust RMM policy, making it an effective solution for real-world indoor localization tasks with class-incremental learning requirements.

The results from Fig. 6 and 7 are more concisely summarized in Fig. 8. CIELO achieved a mean localization error of 0.32 meters and a worst-case error of 0.4 meters, representing up to 29.4× improvement over state-of-the-art methods. In comparison, iCaRL, the next best method, showed a mean error of 0.77 meters, while EWC [16], LwF [18], and FEDHIL [15] demonstrated mean errors up to 9.75 meters and worst-case errors as high as 12.54 meters. The poor performance of these baselines can be attributed to their inherent limitations. EWC's rigid weight constraints restrict adaptation to new RPs, while iCaRL's similarity-based sample retention fails to capture the diversity needed for generalization. LwF struggles with inconsistent task outputs in non-IID environments, as it uses TIL-based training, and FEDHIL lacks an efficient mechanism for incremental learning. CIELO's ability to mitigate these challenges through strategic RM optimization and prioritization of impactful samples establishes it as a robust and efficient solution for CIL in dynamic indoor environments.

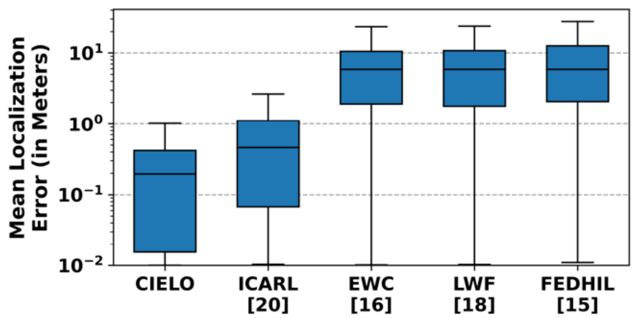


FIGURE 8. Comparison of CIELO against the state-of-the-art.

E. IMPACT OF NEW CLASS INCREMENT SIZE ON PERFORMANCE

To assess adaptability to varying class increments and retraining efficiency, we evaluated mean localization error and retraining loss for increments of 2, 5, and 10 new RPs, again comparing CIELO with the frameworks from iCaRL [20], LwF [18], EWC [16], and FEDHIL [15]. The results are shown in Fig. 9. Across all methods, larger class increments (5 and 10 RPs) resulted in lower localization errors compared to smaller increments (2 RPs). This trend reflects the advantage of fewer retraining cycles with larger increments, which reduce disruptions to previously learned representations and mitigate catastrophic forgetting. CIELO consistently achieved the lowest errors, leveraging memory optimization and sensitivity-based sample selection to retain critical data and maintain generalization. In contrast, baseline methods struggled due to their inability to efficiently adapt to frequent updates and incremental learning in non-IID settings, resulting in higher localization errors.

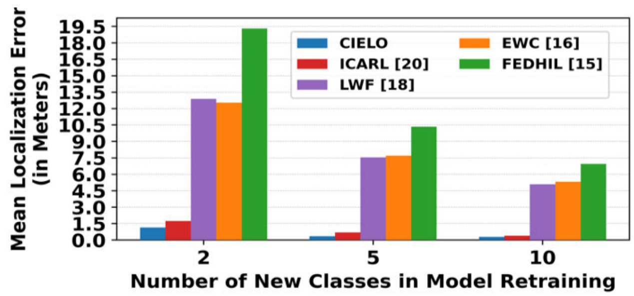


FIGURE 9. Mean localization error with varying number of new classes (RPs) introduced through retraining.

Fig. 10 shows the average retraining loss across all buildings and testing devices for the three scenarios explored in Fig. 9. CIELO demonstrated faster convergence and lower retraining losses by prioritizing impactful samples, effectively mitigating catastrophic forgetting and outperforming other frameworks.

F. EVALUATION OF COMPUTATIONAL COST ON EDGE SERVER

Lastly, the computational cost of each framework was evaluated in terms of average %CPU usage, average power consumption and average training latency during continual retraining on an edge server that is equipped with an AMD Ryzen 7 5800HS processor. Fig. 11 shows the results for the retraining across the scenarios depicted in Fig. 6 and 7, averaged over the two buildings considered. Also shown for reference is the mean localization error for each framework.

It can be observed that CIELO achieves the best balance, with a low mean localization error of 0.32 meters and efficient resource usage (24.92% CPU, 18.72W, 8.82s). In contrast, EWC [16] and FEDHIL [15] exhibited high CPU usage (over 54%) and power consumption (~29W). EWC's fixed weight constraints require frequent computations to preserve prior knowledge, increasing overhead, while FEDHIL's GM aggregation technique involves extensive updates, making it resource intensive. iCaRL [20], with moderate resource usage, benefits from its selective replay mechanism but still incurs costs due to replay computations. LwF [18], while resource-efficient (17.56W, 21.6% CPU, 22.64s), uses task-output preservation, requiring fewer computations but sacrificing accuracy due to disruptions in dynamic environments. CIELO's slight increase in overhead compared to iCaRL [20] and LwF [18] stems from its RMM, which incorporates both confidence and sensitivity measures. This approach carefully selects and adapts key information, rather than relying solely on past data, as in iCaRL, or direct task-output alignment, as in LwF. These results highlight CIELO's ability to reduce edge server resource utilization while maintaining high localization performance, making it suitable for resource-limited and evolving indoor scenarios.

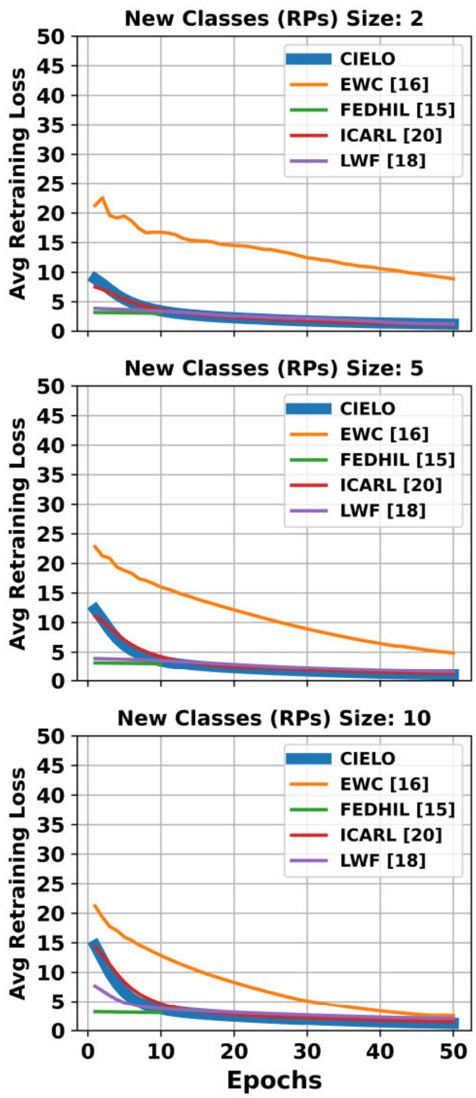


FIGURE 10. Average retraining loss with varying number of new classes (RPs) introduced through retraining.

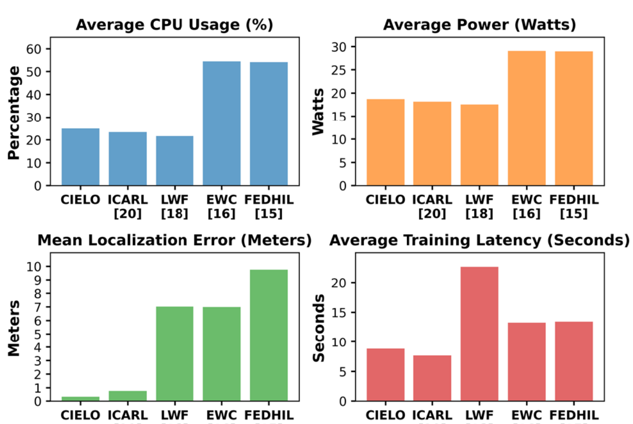


FIGURE 11. Average CPU usage, average power, average training latency, and mean localization error comparison across frameworks.

VI. CONCLUSION

In this paper, we introduced CIELO, a novel Class-Incremental Learning (CIL) framework tailored for Wi-Fi RSS-based indoor localization. CIELO is designed to address the challenges of learning new indoor areas efficiently while mitigating catastrophic forgetting and reducing data storage overhead. To the best of our knowledge, CIELO is the first framework designed to support continual learning and CIL for indoor localization. By leveraging crowdsourced data and implementing a novel representation memory management (RMM) policy, CIELO seamlessly integrates new indoor locations while retaining knowledge of previously learned locations. Experimental results highlight CIELO's effectiveness, demonstrating up to 29.4× improvement in localization accuracy (while learning new classes), up to 1.75× reduction in data storage requirements, and up to 1.69× lower device power consumption compared to state-of-the-art approaches.

REFERENCES

[1] A. Billa, I. Shayea, A. Alhammadi, Q. Abdullah, and M. Roslee, "An overview of indoor localization technologies: Toward IoT navigation services," in *Proc. IEEE 5th Int. Symp. Telecommun. Technol. (ISTT)*, Nov. 2020, pp. 76–81.

[2] *Indoor Location Market by Component (Hardware, Services, Software), Technology (Bluetooth Low Energy (BLE), Infrared, Magnetic Positioning), Organization Size, Application, Deployment, End-User-Global Forecast 2025–2030*. Accessed: Feb. 27, 2025. [Online]. Available: https://www.giiresearch.com/report/ires1580990-indoor-location-market-by-component-hardware.html

[3] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Commun. Surveys Tuts.*, vol. 21, no. 3, pp. 2568–2599, 3rd Quart., 2019.

[4] Y. You and C. Wu, "Indoor positioning system with cellular network assistance based on received signal strength indication of beacon," *IEEE Access*, vol. 8, pp. 6691–6703, 2020.

[5] A. Yassin, Y. Nasser, M. Awad, A. Al-Dubai, R. Liu, C. Yuen, R. Raulefs, and E. Aboutanios, "Recent advances in indoor localization: A survey on theoretical approaches and applications," *IEEE Commun. Surveys Tuts.*, vol. 19, no. 2, pp. 1327–1346, 2nd Quart., 2017.

[6] Z. Yang, Z. Zhou, and Y. Liu, "From RSSI to CSI: Indoor localization via channel response," *ACM Comput. Surveys*, vol. 46, no. 2, pp. 1–32, Nov. 2013.

[7] S. Tiku and S. Pasricha, *Machine Learning for Indoor Localization and Navigation*. Cham, Switzerland: Springer, 2023.

[8] H. Obeidat, W. Shuaieb, O. Obeidat, and R. Abd-Alhameed, "A review of indoor localization techniques and wireless technologies," *Wireless Pers. Commun.*, vol. 119, no. 1, pp. 289–327, Jul. 2021.

[9] G. Pecoraro, S. Di Domenico, E. Cianca, and M. De Sanctis, "CSI-based fingerprinting for indoor localization using LTE signals," *EURASIP J. Adv. Signal Process.*, vol. 2018, no. 1, pp. 1–15, Dec. 2018.

[10] T. Lesort, V. Lomonaco, A. Stoian, D. Maltoni, D. Filliat, and N. Díaz-Rodríguez, "Continual learning for robotics: Definition, framework, learning strategies, opportunities and challenges," *Inf. Fusion*, vol. 58, pp. 52–68, Jun. 2020.

[11] S. Mittal, S. Galesso, and T. Brox, "Essentials for class incremental learning," in *Proc. IEEE/CVF Conf. Comput. Vis. Pattern Recognit. Workshops (CVPRW)*, Jun. 2021, pp. 3508–3517.

[12] G. M. van de Ven, T. Tuytelaars, and A. S. Tolias, "Three types of incremental learning," *Nature Mach. Intell.*, vol. 4, no. 12, pp. 1185–1197, Dec. 2022.

[13] D. R. Ashley, S. Ghiassian, and R. S. Sutton, "Does the Adam optimizer exacerbate catastrophic forgetting?" 2021, *arXiv:2102.07686*.

[14] R. Coop and I. Arel, "Mitigation of catastrophic forgetting in recurrent neural networks using a fixed expansion layer," in *Proc. Int. Joint Conf. Neural Netw. (IJCNN)*, Aug. 2013, pp. 1–7.

[15] D. Gufran and S. Pasricha, "FedHIL: Heterogeneity resilient federated learning for robust indoor localization with mobile devices," *ACM Trans. Embedded Comput. Syst.*, vol. 22, no. 5s, pp. 1–24, Oct. 2023.

[16] J. Kirkpatrick, R. Pascanu, N. Rabinowitz, J. Veness, G. Desjardins, A. A. Rusu, K. Milan, J. Quan, T. Ramalho, A. Grabska-Barwinska, D. Hassabis, C. Clopath, D. Kumaran, and R. Hadsell, "Overcoming catastrophic forgetting in neural networks," *Proc. Nat. Acad. Sci. USA*, vol. 114, no. 13, pp. 3521–3526, Mar. 2017.

[17] M. De Lange, R. Aljundi, M. Masana, S. Parisot, X. Jia, A. Leonardis, G. Slabaugh, and T. Tuytelaars, "A continual learning survey: Defying forgetting in classification tasks," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 44, no. 7, pp. 3366–3385, Jul. 2022.

[18] Z. Li and D. Hoiem, "Learning without forgetting," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 40, no. 12, pp. 2935–2947, Dec. 2018.

[19] F. Potortì et al., "The IPIN 2019 indoor localisation competition–description and results," *IEEE Access*, vol. 8, pp. 206674–206718, 2020.

[20] S.-A. Rebuffi, A. Kolesnikov, G. Sperl, and C. H. Lampert, "ICaRL: Incremental classifier and representation learning," in *Proc. IEEE Conf. Comput. Vis. Pattern Recognit. (CVPR)*, Jul. 2017, pp. 5533–5542.

[21] D. Lymberopoulos and J. Liu, "The Microsoft indoor localization competition: Experiences and lessons learned," *IEEE Signal Process. Mag.*, vol. 34, no. 5, pp. 125–140, Sep. 2017.

[22] G. Chen, X. Guo, K. Liu, X. Li, and J. Yang, "RWKNN: A modified WKNN algorithm specific for the indoor localization problem," *IEEE Sensors J.*, vol. 22, no. 7, pp. 7258–7266, Apr. 2022.

[23] S. Tiku, S. Pasricha, B. Notaros, and Q. Han, "A hidden Markov model based smartphone heterogeneity resilient portable indoor localization framework," *J. Syst. Archit.*, vol. 108, Sep. 2020, Art. no. 101806.

[24] D. D. Nguyen and M. Thuy Le, "Enhanced indoor localization based BLE using Gaussian process regression and improved weighted kNN," *IEEE Access*, vol. 9, pp. 143795–143806, 2021.

[25] P. S. Varma and V. Anand, "Indoor localization for IoT applications: Review, challenges and manual site survey approach," in *Proc. IEEE Bombay Sect. Signature Conf. (IBSSC)*, Nov. 2021, pp. 1–6.

[26] A. B. Adege, H.-P. Lin, G. B. Tarekegn, and S.-S. Jeng, "Applying deep neural network (DNN) for robust indoor localization in multi-building environment," *Appl. Sci.*, vol. 8, no. 7, p. 1062, Jun. 2018.

[27] X. Song, X. Fan, X. He, C. Xiang, and Z. Wang, "CNNLoc: Deep-learning based indoor localization with WiFi fingerprinting," in *Proc. IEEE SmartWorld, Ubiquitous Intell. Comput., Adv. Trusted Comput., Scalable Comput. Commun., Cloud Big Data Comput., Internet People Smart City Innov. (SmartWorld/SCALCOM/UIC/ATC/CBDCom/IOP/SCI)*, Aug. 2019, pp. 589–595.

[28] S. Tiku, D. Gufran, and S. Pasricha, "Multi-head attention neural network for smartphone invariant indoor localization," in *Proc. IEEE 12th Int. Conf. Indoor Positioning Indoor Navigat. (IPIN)*, Sep. 2022, pp. 1–8.

[29] P. Blanchard, EME. Mhamdi, R. Guerraoui, and J. Stainer, "Machine learning with adversaries: Byzantine tolerant gradient descent," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–12.

[30] F. Yin, Z. Lin, Q. Kong, Y. Xu, D. Li, S. Theodoridis, and S. R. Cui, "FedLoc: Federated learning framework for data-driven cooperative localization and location data processing," *IEEE Open J. Signal Process.*, vol. 1, pp. 187–215, 2020.

[31] A. A. Rusu, N. C. Rabinowitz, G. Desjardins, H. Soyer, J. Kirkpatrick, K. Kavukcuoglu, R. Pascanu, and R. Hadsell, "Progressive neural networks," 2016, *arXiv:1606.04671*.

[32] H. Shin, J. K. Lee, J. Kim, and J. Kim, "Continual learning with deep generative replay," in *Proc. Adv. Neural Inf. Process. Syst.*, 2017, pp. 1–11.

[33] J. X and Z. Zhanxing, "Reinforced continual learning," in *Proc. Adv. Neural Inf. Process. Syst.*, 2018, pp. 1–13.

[34] C. Lamers, R. Vidal, N. Belbachir, N. van Stein, T. Bäck, and P. Giampouras, "Clustering-based domain-incremental learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2023, pp. 3376–3384.

[35] P. Garg, R. Saluja, V. N. Balasubramanian, C. Arora, A. Subramanian, and C. V. Jawahar, "Multi-domain incremental learning for semantic segmentation," in *Proc. IEEE/CVF Winter Conf. Appl. Comput. Vis. (WACV)*, Jan. 2022, pp. 2080–2090.

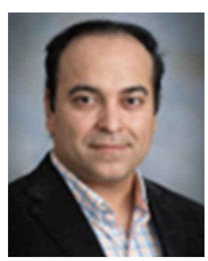
[36] G. Oren and L. Wolf, "In defense of the learning without forgetting for task incremental learning," in *Proc. IEEE/CVF Int. Conf. Comput. Vis. Workshops (ICCVW)*, Oct. 2021, pp. 2209–2218.

[37] DW. Zhou, QW. Wang, ZH. Qi, HJ. Ye, DC. Zhan, and Z. Liu, "Class-incremental learning: A survey," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 46, no. 12, pp. 9851–9873, Dec. 2024.

[38] A. Abolfazli and E. Ntoutsi, "Drift-aware multi-memory model for imbalanced data streams," in *Proc. IEEE Int. Conf. Big Data (Big Data)*, Dec. 2020, pp. 878–885.

AKHIL SINGAMPALLI (Graduate Student Member, IEEE) received the B.Tech. degree in electronics and communication engineering from the Vignan's Institute of Information Technology, India. He is currently pursuing the Ph.D. degree in computer engineering with Colorado State University, Fort Collins. His research interests include continual and federated machine learning, with a focus on developing robust, secure, and real-time solutions for indoor localization using mobile and IoT devices.

DANISH GUFRAN (Member, IEEE) received the B.E. degree in electronics and instrumentation engineering from Visvesvaraya Technological University, India, in 2020. He is currently pursuing the Ph.D. degree in computer engineering with Colorado State University, Fort Collins. His contributions have been published in various IEEE and ACM conferences and journals. His research focuses on designing innovative low-power deep machine learning algorithms to develop energy-efficient models suitable for edge and IoT devices. Additionally, his work extends to implementing federated learning systems and adversarial security models for indoor localization. He is a member of ACM and Sigma Xi. He was a recipient of the best paper award and has co-authored multiple book chapters.



SUDEEP PASRICHA (Fellow, IEEE) received the Ph.D. degree in computer science from the University of California at Irvine, in 2008. He is currently a Walter Scott Jr. College of Engineering Professor with the Department of Electrical and Computer Engineering, Colorado State University. His research focuses on the design of innovative software algorithms, hardware architectures, and hardware–software co-design techniques for energy-efficient, fault-tolerant, real-time, and secure computing. He is a fellow of AIAA, a Distinguished Member of ACM, and an ACM Distinguished Speaker. He has co-authored five books and published more than 300 research articles in peer-reviewed journals and conferences that have received 17 best paper awards and nominations at various IEEE and ACM conferences. He has served as the general chair and the program committee chair for multiple IEEE and ACM conferences and served in the editorial board of multiple IEEE and ACM journals.