Novel Fingerprint-based inference framework based on Generative Adversarial Network

Abstract—**As indoor localization becomes a necessity to provide intelligent location-based services for in-building users, fingerprint-based positioning has been widely adopted in a number of Wi-Fi-equipped devices. However, this approach requires an extensive prior site survey and thus can not be applied to unexplored environments where any prior fingerprint sampling has not been conducted. To address the problem, we propose a novel fingerprint-based inference framework based on a Generative Adversarial Network (GAN) by extracting the underlying correlation between a location coordinate and its radio signal features. This work empowers indoor localization in unknown areas, including unknown data points, newly deployed APs, or unexplored sites, by 1) decomposing into a signal feature map for each AP; 2) processing learning with a set of location and its associated signal strength; 3) generating and integrating synthetic radio fingerprints; and 4) employing them into some existing localization algorithms. We evaluated GAN- Loc with extensive real-world RSSI experiments in seven different real-world indoor places across various wireless radios.**

We propose three problems in this project. The first one is to set up two different APs in one known place. We randomly walk in the room and get different Received Signal Strength in different location to two separate APs. Then we use GAN to generate the localization fingerprint map of the whole room. The second one is to set up one APs in one known room and also get different RSS in our random path. Then we use GAN to generate different fingerprint maps from different APs. The third one is we get the RSS from a known area and its fingerprint map from one settles AP and generate other fingerprint map in other unknown area.

Index Terms—**Fingerprint database, GAN, RSS**

I. Introduction

As demand for localization services surges, fingerprint-based indoor localization technology has become the predominant method for positioning due to its exceptional precision and minimal hardware requirements. This technology is distinguished not only by its accuracy but also by its low complexity and the minimal processing time required, making it well-suited for mobile devices. Fingerprint-based indoor localization operates effectively within these constraints, relying on the collection of Received Signal Strength (RSS) or Channel State Information (CSI) from surrounding access points[1]. This data is essential for constructing a comprehensive fingerprint database at each reference point. To ensure the efficacy and responsiveness of this technology, it is crucial to implement robust mechanisms for data collection, database management, and algorithm optimization to maintain high performance standards in dynamic indoor environments.

The rising popularity of mobile and pervasive computing has driven extensive research in wireless indoor localization, facilitating the provision of room-level location-based services, such as locating individuals or devices within buildings. However, the received signal strength (RSS) or channel state information (CSI) from surrounding access points must be measured at each reference point to build a fingerprint database [1].

Predominantly, existing localization approaches leverage Received Signal Strength (RSS) as a fundamental metric for determining positions. RSS fingerprints, which are readily obtainable from standard wireless network equipment like Wi-Fi or ZigBee devices, form the basis of these methods. Localization typically unfolds in two phases: offline phase and online phase [2]. Initially, the offline phase involves a site survey where RSS fingerprints are collected at various points within a designated area to create a comprehensive fingerprint database. Each fingerprint is associated with its specific location.

Subsequently, in the online phase, when a user submits a location query along with their current RSS fingerprint, localization algorithms consult the database to identify and return the closest matching fingerprints and their corresponding locations. This phase involves real-time data recording by a mobile device, which then compares this data against the pre-established database. The location of the device is assumed to be that of the closest matching reference point.

While many modern systems employ Channel State Information (CSI) for fingerprints, owing to its enhanced precision and reliability, RSS remains preferable for scenarios that are resource-constrained or sensitive to costs due to its simplicity and minimal hardware requirements. Systems such as Horus utilize probabilistic methods to estimate locations using RSS data [3], capitalizing on the ease of obtaining and processing RSS measurements, which are also typically faster to acquire in real-time, beneficial for applications requiring immediate performance, such as signal strength indicators in mobile communications.

Despite its advantages, the RSS-based method's dependence on extensive site surveys—which are labor-intensive, time-consuming, and susceptible to environmental changes—poses significant challenges. These surveys are crucial as they ensure the fingerprint database accurately reflects the on-site signal landscape, yet the scalability of such systems is often limited by the availability of comprehensive interior fingerprint data.

Furthermore, advancements in wireless and embedded technologies have significantly enriched the smartphone market, equipping these devices with robust computational and sensory capabilities. Constantly carried by users, smartphones have become vital interfaces between individuals and their surroundings, laying a solid foundation for breakthroughs in indoor localization technology. These developments promise to enhance the precision and utility of localization systems, making them increasingly integral to our digital and physical worlds.

Building upon existing research, this study reassesses current localization schemes and explores utilizing previously untapped data sources. We consider user movements within a building, linking geographically separate Received Signal Strength (RSS) fingerprints through user movement paths. This method creates a high-dimensional fingerprint space where the distances among fingerprints, measured in footsteps, are preserved. Additionally, we redefine the building's floor plan into a "stress-free floor plan"—a high-dimensional space where the walking distance between two points reflects their actual physical distance in the real floor plan. This spatial alignment between the stress-free floor plan and the fingerprint space facilitates the labeling of fingerprints with actual locations, a task traditionally achieved through site surveys. These insights drive our proposal for practical, flexible, and rapidly deployable localization approaches that require minimal human effort and intervention.

A significant challenge in fingerprint-based localization is determining the requisite data volume to achieve desired accuracy during the offline phase. Subsequent sections demonstrate that a larger fingerprint database enhances localization performance. However, assembling a comprehensive database is both time-consuming and labor-intensive [1]. For example, our experiments show that collecting fingerprint data at a single reference point consumes approximately thirty minutes. This extensive time requirement hampers the widespread adoption and practical application of fingerprint-based localization. To mitigate data collection costs, several strategies have been suggested [1], [4], [5]. For instance, one method employs compressive sensing to reconstruct missing fingerprints, revealing the underlying structure and redundancy in fingerprints through a merging matrix [1]. Another approach uses semi-supervised manifold learning to build a fingerprint database from partially labeled data, reducing the need for fully labeled signal strength measurements [4].

To further enhance the efficiency of building the fingerprint database while minimizing human labor, we introduce a novel method using generative adversarial networks (GANs) [6], [7]. This technique involves transforming RSS collected at each reference point into amplitude feature maps. These maps are then converted into high-resolution images through pixel transformation, forming the basis of the initial fingerprint database. By mapping the amplitude feature maps of the reference points, this database inherently contains detailed location information for all reference points, facilitating a more streamlined and efficient database construction process.

Finally, the generated amplitude feature maps are merged into the initial fingerprint database, resulting in an expanded database. As described in the following section, the localization accuracy improves as the number of samples in the fingerprint database increases. We also evaluate the proposed fingerprint construction method through extensive experiments in a typical indoor classroom environment.

The main contributions of this paper are as follows:

1) We build a fingerprint database by converting processed RSS data into amplitude feature maps. This approach visualizes the position of the sampling points and allows us to visually determine the locations of the testing points.

2) Based on the nature of the amplitude feature maps, a GAN model is proposed that converges quickly and generates samples with improved diversity. We use the GAN model to generate additional amplitude feature maps of the sampling points' positions, which reduces the collection time associated with each single sample point and saves human effort.

3) Two experiments are conducted and the results show that GAN with BCE loss function can provide better performance compared to the standard GAN.

II. Related Work

A. Wireless Localization

Over the past two decades, indoor localization techniques have notably evolved and can be broadly categorized into two types: fingerprinting-based and model-based approaches. A predominant method in indoor localization is fingerprint matching, which involves capturing unique environmental signatures at various locations within areas of interest to create a comprehensive fingerprint database. These fingerprints are then used to estimate locations by matching real-time data against the stored fingerprints. Researchers have explored utilizing signatures from commonly available devices and reducing the mapping efforts. Techniques typically harness RF signals, as exemplified by systems like RADAR [8], Horus [3], LANDMARC [9], ActiveCampus [10], PlaceLab [11], and OIL [12]. Additionally, Surround Sense [13] estimates locations based on ambient features such as sound, light, color, and WiFi signals. Recent innovations have included using FM radio [14] and Channel Frequency Response [15] as fingerprinting sources. Despite their efficacy, these methods often require extensive site surveys to build databases, incurring significant manual effort and costs while lacking flexibility to adapt to environmental changes.

Contrary to fingerprinting, model-based approaches determine locations using geometrical models that predict RF propagation distances from measured RSS values, as seen in the log-distance path loss (LDPL) model [16]. These techniques generally sacrifice some localization accuracy to reduce the measurement effort required.

B. Simultaneous Localization and Mapping (SLAM)

In robotics and computer vision, SLAM techniques enable the concurrent estimation of both a robot's location and a map of the environment. However, the variable nature of wireless signal strength complicates the application of traditional SLAM methods, which typically rely on detecting distinct landmarks or navigating precisely controlled robot movements [17]. Our proposed solution avoids using a digital compass or gyroscope; instead, it employs an accelerometer as a pedometer to record the number of user footsteps, accurately measuring user displacement and direction with today's smartphones [18]. The deterministic Multidimensional Scaling (MDS) method calculates locations, though the correlation between the discovered and actual world mappings remains underexplored in SLAM literature and our approach.

C. Multidimensional Scaling (MDS)

Multidimensional scaling (MDS) [19] is a statistical technique used to visualize the similarities or dissimilarities in data by positioning items in a predetermined dimensional space (typically 2D or 3D for visualization purposes). In network localization, MDS is instrumental in assigning coordinates to nodes such that the distances measured to neighboring nodes (via RSS, ToA, TDoA, etc.) are preserved. While our use of MDS shares similarities with these network applications, it uniquely focuses on user localization without relying on fingerprinting techniques.

III. Model Architecture and Approach

A. Process

- Decomposition

  Using participation is essential in the initial period at the online stage. Untrained users walk in a building following daily activities. Mobile phones, carried by the users, collect Wi-Fi RSS fingerprints at various locations along user movement paths, and the walking distances are also recorded. Walking distances accelerometers in mobile phones. Similarly, accelerometers also infer the starting and finishing points movements of the user paths. Also, due to the limitation of the site survey and data collection, we need to place several APs rather than only one AP in the same test area, which can help provide sufficient training data to the generator. Then, we need to separate the data coming from the different APs and classify them into different training datasets.
- Learning

  Using different datasets from the same APs and same area to train the generator model to learn the connection between access point coordinates and received signal strength in different coordinates. Then we use discriminator to discriminate the generated data from the true data we collected, which helps the generator and the discriminator both learn and reaches the perfect status.
- Generation

  After using adequate data for generator to learn the connection, we apply other coordinates to the generator to generate the associated received signal strength that are difficult for us to collect.
- Localization

  After the generator and the discriminator are trained for enough times to the best convergence, the localization fingerprint database is fully constructed. When a location query comes, usually an RSS fingerprint sent by a user, we take it as a keyword and searches the fingerprint database. We can then use the map to search the most matched localization with this map by using the simplest searching algorithm to see which coordinate has the closest received signal strength in the map, then the location will be settled.

B. Three Problems Proposed

- Problem One

  Situated within a predefined, known area, the first problem entails strategically placing various access points (APs) at different coordinates. The received signal strengths (RSS) from these coordinates are collected, which are then utilized to facilitate the training of the discriminator. This training process helps the generator to effectively learn the relationship between the locations of the APs and the RSS values received from other locations. The primary goal here is to enhance the accuracy of the localization system by refining the generator's ability to predict RSS based on the AP placement.

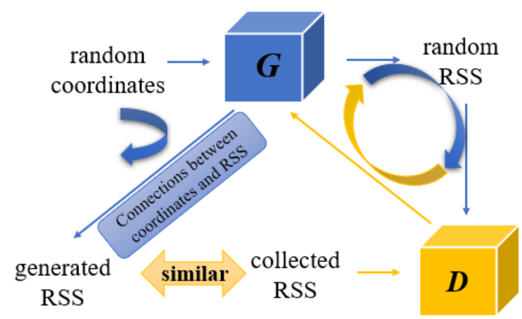

Fig. 1. Problem One

- Problem Two

  Continuing in the same known area, the second problem involves an initial collection of data from different APs positioned at various coordinates. The objective here is to generate the RSS as it would appear if the signal were emanating from alternate coordinates relative to the APs. To accomplish this, the discriminator is fed the collected RSS data, and the generator receives both the coordinates of the APs and a set of randomly selected coordinates. This configuration is designed to train the generator to establish a robust connection between the AP coordinates and the corresponding RSS, enhancing the system's versatility in real-world applications.
- Problem Three

  Expanding the study's scope, the third problem addresses different geographical areas. It starts with the collection of RSS data from various APs spread across different regions, which is then organized into a comprehensive database. In this scenario, the significance of the map matrix is emphasized. The generator is provided with the area map, the coordinates of the APs, and the random coordinates from which the RSS was originally collected.

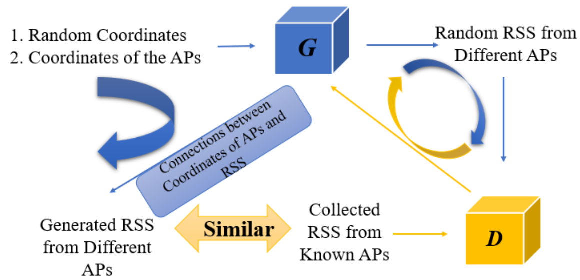

Fig. 2. Problem Two

Subsequently, the discriminator is supplied with the RSS data from these diverse areas. The training process is meticulously designed to help the generator learn and accurately map the intricate relationships among the map distribution matrix, the AP coordinates, and the RSS from various locations.

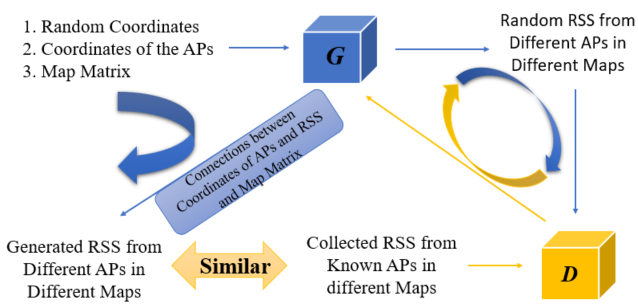

Fig. 3. Problem Three

C. Stress-Free Floor Plan

A floor plan provides an overhead view of a building's layout, depicting the relationships between rooms, spaces, and other physical features. However, the geographical distance between two points on a floor plan may not correspond to the actual walking distance due to obstructions such as walls and other barriers. To address this discrepancy, we propose the concept of a stress-free floor, which utilizes Multidimensional Scaling (MDS) to project real locations into a higher-dimensional space. In this space, the geometric distances between points more accurately reflect the true walking distances, allowing for a more effective utilization of walking data collected by users.

In the fingerprint space, it constitutes a unique aspect, diverging from traditional methodologies. Here, MDS is employed to generate a high-dimensional space where fingerprints are spatially unrelated, thus eliminating the conventional possibility of creating a coherent fingerprint space based on geographical proximity.

In the fingerprint database, fingerprints are directly associated with the locations where they were collected, effectively labeling each fingerprint with its specific location. This association is facilitated by mapping the fingerprint space to the stress-free floor plan, thereby aligning each fingerprint with its accurate geographical position in the building.

Furthermore, we use a systematic grid-based matrix approach to transform the spatial features of the map into a matrix representation of the map. For example, the interior space is divided into grids, each translated into a matrix where obstructions are indicated by -1, unreachable areas by 0, and accessible, obstruction-free zones by 1, as shown in the Figure 4 and 5.

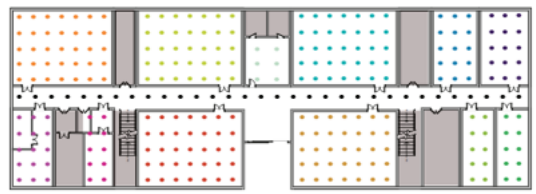

Fig. 4. Stress-Free Floor Plan

1	1	1	-1	1	1	-1	-1	1	1	1	-1	1	1
1	1	1	-1	1	1	1	1	1	1	1	-1	1	1
1	1	1	-1	1	1	1	1	1	1	1	-1	1	1
1	1	1	1	1	1	1	1	1	1	1	1	1	1
1	-1	1	-1	1	1	0	0	1	1	1	-1	-1	1
1	-1	1	-1	1	1	0	0	1	1	1	-1	-1	1

Fig. 5. Map Matrix

D. Generative Adversarial Network

Generative Adversarial Networks (GANs) are a type of generative model that employs a game-theoretic approach to conduct adversarial training between a generative model and a discriminative model. This training process enables the generation of synthetic data such as images and audio. The core concept of GANs involves two networks—the generator and the discriminator—engaging in a strategic game. Through the optimization of this adversarial process, the generator progressively improves, the ultimate goal is to iteratively develop a high-quality generator.

- Generator

  The primary function of the generator in Generative Adversarial Networks (GANs) is to take random noise as input and output synthetic data. This random noise is a vector following a Gaussian distribution, which is then transformed into synthetic data through a deep neural network model, such as a Convolutional Neural Network (CNN) or a Feedforward Neural Network.

- Discriminator

  The discriminator is a binary classification neural network model, receiving inputs either from the generator or real data. Its task is to classify the input data and output a probability value indicating whether the data originates from the real dataset or is synthetic data generated by the generator.
- Adversarial Process

  The adversarial process between the generator and the discriminator constitutes their respective training processes. The training objective for the generator is to maximize the probability that the discriminator classifies its output as real data. Conversely, the discriminator aims to enhance its accuracy in distinguishing between real and generated data. Through iterative training, the generator progressively becomes adept at producing synthetic data that closely mimics real data.

  Suppose $P_{\text{data}}(x)$ represents the distribution of real data, $P_{\text{z}}(z)$ denotes the distribution of the random noise z input into the generator, and $G(z;\theta_g)$ symbolizes the output of the generator, where $P_{\text{z}}(z)$ is the parameter of the generator. Similarly, $D(x;\theta_d)$ represents the output of the discriminator, and θ_d refers to the parameters of the discriminator.

  In the context of GANs, the objective is to minimize the following loss function:

$$\min_G \max_D V(D,G) = \mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)] + \mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))] \tag{1}$$

  This loss function aims to minimize the discrepancy between the synthetic data produced by the generator and the real data, while maximizing the discriminator's ability to differentiate between the two. Specifically, the first term $\mathbb{E}_{x\sim p_{\text{data}}(x)}[\log D(x)]$ represents the probability that real data is correctly identified as real by the discriminator, while the second term $\mathbb{E}_{z\sim p_z(z)}[\log(1-D(G(z)))]$ represents the probability that synthetic data generated by the generator is correctly identified as such.

  During training, GANs alternately train the generator and the discriminator, optimizing model parameters by minimizing the loss function. Typically, each training iteration involves first fixing the parameters of the generator and optimizing those of the discriminator by maximizing the loss function $V(D,G)$. Subsequently, the discriminator's parameters are fixed, and the generator's parameters are optimized by minimizing the loss function $V(D,G)$. This iterative process continues until a predetermined number of iterations is reached or the loss function $V(D,G)$ converges.

E. Model of Problem One

In the first problem, we employ a multilayer perceptron, or fully connected neural network, as the generator. The inputs to the generator are the information about the signal source and the coordinates on the map, while the outputs are the corresponding signal strengths at these map coordinates.

Then a model comprising a fully connected neural network serves as the discriminator. The inputs to the discriminator are the real signal strengths at various coordinates under a specified signal source from the training set and the pseudo signal strengths generated by the generator. The outputs of the discriminator are the probabilities that assess the authenticity of these signal strengths.

The Generative Adversarial Network (GAN) alternates between training the generator and the discriminator, which are both composed of fully connected networks. Through numerous iterations, as the loss function converges or the number of iterations increases, the signal strengths generated by the generator—based on the information of the signal source and coordinates—increasingly approximate the actual signal strengths. Concurrently, the discriminator's accuracy in distinguishing whether the signal strengths at given coordinates under specified signal source conditions are real or generated by the generator improves. Upon completion of the training, the generator attains the capability to generate accurate signal strengths based on the signal source coordinates and the coordinates of the required signal strength locations. In other words, the generator's fully connected neural network can adequately fit the distribution rules of the signal source coordinates, the map point coordinates, and their corresponding signal strengths. At this stage, the generator can be isolated and used to predict the signal strengths at other coordinates under specified signal source conditions.

The discriminator in this task utilizes binary cross-entropy as its loss function, which effectively quantifies the accuracy of its binary classification:

$$L = -\sum_{i=1}^{N}[y_i\log(\hat{y}_i) + (1-y_i)\log(1-\hat{y}_i)] \tag{2}$$

In the model, $y_i$represents the probability distribution of the actual signal strength at the i-th coordinate, while $\hat{y}_i$ denotes the probability distribution of the synthetic signal strength produced by the generator at the same coordinate.

To accelerate model convergence and minimize error, the discrepancy $loss_p$ between the synthetic and actual signal strengths is also incorporated as part of the loss function for training the generator. Experimental results indicate that this approach effectively speeds up model convergence and enhances the reliability of the predictions.

IV. Experiment

A. Experimental Design and Settings

In terms of experimental design, we selected 50 datasets, each representing signal intensities at various locations within the same environment, emanating from signal sources of different intensities. Each dataset comprises coordinates and corresponding signal strengths for 49 points, including the signal source. The experiments were conducted in two sets: the first set involved randomly selecting 41 points for training

Algorithm 1 Training GAN with Binary Cross-Entropy Loss

Initialize generator G and discriminator D and Binary Cross-Entropy Loss Function $BCEloss$

Inputs: coordinates (x, y) and signal strength P of the signal source, coordinates (x, y) and signal strength p

Outputs: predicted signal strength p', and the true probability of signal strength t.

```
for i in epochs do
    Reset discriminator parameter gradients to zero
    P' = G((x, y), P, x, y, p)
    t = D((x, y), P, p')
    D_loss = BCEloss(t)
    D_loss.backward()
    loss_real = BCEloss(D(x_i), 1)
    Reset generator parameter gradients to zero
    P' = G((x, y), P, x, y, p)
    t_1 = D((x, y), P, p')
    t_2 = D((x, y), P, p)
    G_loss = t_1 + t_2 + loss(p, p')
    G_loss.backward()
end for
```

and 8 for testing from each dataset; the second set involved training on 33 randomly selected points and testing on 16 from each dataset. Each set underwent training cycles of 50, 70, and 100 epochs, respectively.

Regarding model parameters, the generator consists of four fully connected layers with configurations set at 5×32, 32×63, 64×32, and 32×1; the discriminator is composed of three fully connected layers with settings at 5×32, 32×16, and 16×1.

Experimental Setup: the experiments were conducted on a system equipped with a 3rd Generation Intel® Core™ i7-9750H CPU and an NVIDIA GeForce RTX 1660 Laptop GPU with 6GB of memory.

B. Performance

The experimental results were evaluated using the discrepancies between the predicted and actual signal strengths across all test coordinates from 50 datasets. Specifically, the model's predictive accuracy for untrained coordinate points was assessed using the mean error across all test points. The orange line represents the predictive performance of the GAN model across different training epochs, while the blue line illustrates the outcomes of the GAN* model, which optimizes the generator using the differences between predictions and actual values as the loss function.

For each variant, ten models were trained per training epoch, with the final results being the average across these ten models. In the first experimental set, as shown in the Figure 6, 8 test points were chosen per dataset, totaling 400 test points across the 50 datasets. For GAN, the average error for these points decreased as the number of training epochs increased, reaching a minimum average error of 4.1516 after 100 training rounds. The GAN* model converged faster, achieving a minimum error of 3.6730 after just 70 rounds.

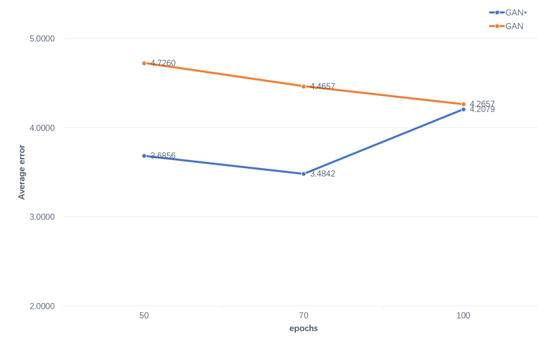

Fig. 6. 8 Spots Generated

In the second experimental set, as shown in the Figure 6, each dataset included 16 test points, culminating in 800 test points across all datasets. For GAN, the average error decreased with additional training epochs, recording a minimum average error of 4.2657 after 100 rounds. Conversely, the GAN* model exhibited a faster convergence, reaching a minimal error of 3.4842 after 70 rounds.

Fig. 7. 16 spots Generated

The results demonstrate that the GAN* model converges faster than the standard GAN and consistently outperforms it in both experimental sets, indicating its enhanced capability to accurately predict signal strengths for untrained coordinate points. This superior performance highlights the effectiveness of using the discrepancy between predicted results and actual values as a criterion for optimizing the generator in the GAN* model.

V. Conclusion

In this paper, we employ a Generative Adversarial Network (GAN) model to generate additional fingerprints that closely mimic those in the training dataset. We conducted two experiments to assess the performance of the proposed method. The results highlighted the effectiveness of GAN, particularly when augmented with an additional loss function, suggesting that this approach could enhance the accuracy of indoor positioning systems without necessitating an increase in the labor required to build extensive fingerprint databases. However, the method does have its limitations, notably that the accuracy of positioning correlates directly with the volume of data in the fingerprint database.

Looking forward, we foresee that localization technology will play an essential role in future societies by enabling a range of enriched applications. To further enhance the accuracy of localization, we plan to explore additional optimization methods. For instance, deep convolutional neural networks will be examined in addressing the next two problems. Furthermore, the GAN-based model will be benchmarked against other popular data generation methods, such as Synthetic Minority Over-sampling Technique (SMOTE) [20], to compare accuracy and computational complexity. This comparison will help identify the most efficient strategies for improving indoor positioning systems.

REFERENCES

[1] Z. Gu, Z. Chen, Y. Zhang, Y. Zhu, M. Lu, and A. Chen, "Reducing fingerprint collection for indoor localization," Comput. Commun., vol. 83, pp. 56–63, 2015.

[2] Q. Li et al., "RMDS: Ranging and multidimensional scaling-based anchorfree localization in large-scale wireless sensor networks with coverage holes," Int. J. Distrib. Sensor Netw., vol. 13, pp. 1–12, 08 2017.

[3] M. Youssef and A. Agrawala, "The Horus WLAN location determination system," in Proc. 3rd Int. Conf. Mobile Syst., Appl., and Serv., New York, NY, USA, 2005, pp. 205–218.

[4] T. Pulkkinen, T. Roos, and P. Myllymäki, "Semi-supervised learning for WLAN positioning," in Proc. 21th Int. Conf. Artif. Neural Netw., Berlin, Heidelberg, 2011, pp. 355–362.

[5] S. Liu, H. Luo, and S. Zou, "A low-cost and accurate indoor localization algorithm using label propagation based semi-supervised learning," in Proc. 5th Int. Conf. Mobile Ad-hoc Sensor Netw., Dec. 2009, pp. 108–111.

[6] I. Goodfellow et al., "Generative adversarial nets," in Proc. Advances Neural Inf. Process. Syst., 2014, pp. 2672–2680.

[7] Y.-J. Cao et al., "Recent advances of generative adversarial networks in computer vision," IEEE Access, vol. 7, pp. 14985–15006, 2019.

[8] P. Bahl and V. N. Padmanabhan. RADAR: an in-building RF-based user location and tracking system. In Proceedings of IEEE INFOCOM, volume 2, pages 775–784, 2000..

[9] L. M. Ni, Y. Liu, Y. C. Lau, and A. P. Patil. LANDMARC: indoor location sensing using active RFID. Wireless Networks, 10(6):701–710, 2004.

[10] W. G. Griswold, P. Shanahan, S. W. Brown, R. Boyer, M. Ratto, R. B. Shapiro, and T. M. Truong. ActiveCampus: experiments in community-oriented ubiquitous computing. Computer, 37(10):73–81, 2004.

[11] A. LaMarca, Y. Chawathe, S. Consolvo, J. Hightower, I. Smith, J. Scott, T. Sohn, J. Howard, J. Hughes, F. Potter, et al. Place lab: Device positioning using radio beacons in the wild. Pervasive Computing, pages 301–306, 2005.

[12] J. Park, B. Charrow, D. Curtis, J. Battat, E. Minkov, et al. Growing an organic indoor location system. In Proceedings of ACM MobiSys, pages 271–284, 2010.

[13] M. Azizyan, I. Constandache, and R. Roy Choudhury. Surroundsense: mobile phone localization via ambience fingerprinting. In Proceedings of ACM MobiCom, pages 261–272, 2009.

[14] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha. Fm-based indoor localization. In Proceedings of the ACM MobiSys 2012, MobiSys '12, pages 169–182, New York, NY, USA, 2012. ACM.

[15] S. Sen, B. Radunovic, R. R. Choudhury, and T. Minka. You are facing the mona lisa: spot localization using phy layer information. In Proceedings of the ACM MobiSys 2012, MobiSys '12, pages 183–196, New York, NY, USA, 2012. ACM.

[16] H. Lim, L. C. Kung, J. C. Hou, and H. Luo. Zero-configuration indoor localization over IEEE 802.11 wireless infrastructure. Wireless Networks, 16(2):405–420, 2010.

[17] S. Thrun, W. Burgard, and D. Fox. Probabilistic Robotics. MIT Press, 2005.

[18] H. Wang, S. Sen, A. Elgohary, M. Farid, M. Youssef, and R. Choudhury. No need to war-drive: Unsupervised indoor localization. In Proceedings of ACM MobiSys, 2012.

[19] I. Borg and P. Groenen. Modern multidimensional scaling: Theory and applications. Springer Verlag, 2005.

[20] C. Nitesh, B. Kevin, O. Lawrence, and P. Kegelmeyer, "SMOTE: Synthetic minority over-sampling technique," J. Artif. Intell. Res., vol. 16, pp. 321–357, 2002.