WiFi CSI/RSSI Fingerprints Positioning based on Data Augmentation Technique

Mohamed Amin Elaoud and Wiem Fekih Hassen

Chair of Distributed Information Systems, University of Passau, Innstraße 41, 94032 Passau, Germany
Email: {firstname.lastname}@uni-passau.de

Abstract—**In today's digitally connected world, Indoor Positioning Systems (IPS) are of paramount importance, especially for applications within enclosed spaces such as buildings. Leveraging the widespread deployment of WiFi technology, this paper presents an IPS that hinges on WiFi signal data specifically, Received Signal Strength Indicator (RSSI) and Channel State Information (CSI), and the powerful generative model, Tabular Generative Adversarial Network (TabGAN). The work entails a meticulous data collection process conducted within a controlled laboratory environment at the University of Passau, in Germany. Subsequently, data augmentation through GAN is employed to enrich the dataset. The augmented data is then evaluated using LightGBM and Convolutional Neural Network (CNN) models, with the Root Mean Square Error (RMSE) as the primary metric and Positioning Error for comprehensive evaluation of the IPS's accuracy and positioning capabilities. The IPS achieved a remarkable result of 0.99 meters for LightGBM and 0.8 meters for CNN, showcasing its high accuracy on unseen data and validating the efficacy of GAN-based data augmentation for enhancing indoor positioning capabilities.**

Keywords—**IPS, CSI, RSSI, private dataset, Raspberry Pi, GAN, CNN.**

I. Introduction

Localization, the process of determining the locations of entities, devices, and other objects, has become an active research field in recent years. Much of the research focuses on using established technology to determine positions. Depending on the context in which positioning occurs, it can be categorized into two types: outdoor positioning and indoor positioning [1], [2]. While outdoor positioning using Global Navigation Satellite Systems (GNSS) technology, such as the Global Positioning System (GPS), is widely adopted due to the convenience of requiring only one receiver to obtain a position, it fails inside buildings due to signal obstruction by walls and other obstacles. As a result, Indoor Positioning Systems (IPS) have emerged as increasingly essential, particularly for locating people or objects where GPS and other satellite technologies lack precision or fail entirely, such as in hospitals, airports, and underground locations.

Compared to outdoor positioning which usually relies on GPS, indoor positioning doesn't have a one-size-fits-all method. This is mainly due to the unique and varied nature of indoor environments. Instead, any available wireless technique can be utilized to help determine a device's location indoors. Different technologies exist for indoor positioning, and many of them use existing wireless networks, which helps to avoid the need for extra equipment [1].

Various solutions have been proposed for indoor positioning systems. These include technologies that use Bluetooth, WiFi, and Ultra-WideBand (UWB) [3]. One of the most used technologies is WiFi, as it's already found nearly everywhere and can be set up quite easily.

Among different localization technologies, WiFi has gained substantial traction for indoor positioning due to the ubiquitous presence of WiFi-enabled devices and easy access to WiFi Access Points (APs). WiFi-based indoor positioning employs unique mathematical methods or positioning techniques to estimate the location of a device [3]. These techniques include proximity, trilateration, and WiFi fingerprinting [3], [4]. WiFi fingerprinting, in particular, has proven to be an effective and cost-efficient approach for indoor localization [1].

WiFi fingerprinting involves two phases: the offline phase and the online phase. During the offline phase, fingerprints representing Received Signal Strength Indicator (RSSI) or Channel State Information (CSI) measurements are collected at predefined Reference Points (RPs). These fingerprints are stored in a database, called a Radio Map, and are used to train a learning algorithm that maps each fingerprint to its location. In the online phase, the learned model predicts the position of a device based on its RSSI (or CSI) data. However, WiFi fingerprinting faces challenges due to signal fluctuations caused by the multi-path effect and physical obstacles.

Most of Previous research papers have utilized a private dataset [5], [6] and they have achieved a positioning error greater than 1.25 meters. As of the time of writing, there is no publicly available dataset that combines CSI amplitude and phase information with corresponding RSSI values, along with crucial data on collection positions. This comprehensive dataset is essential for training supervised Machine Learning (ML) models effectively. A challenging problem which arises in this domain is the small size of the radio map.

Our research aims at the creation of a CSI (i.e., with amplitude and phase information) and RSSI dataset, the implementation of data augmentation techniques to increase the amount of data and the application of Deep Learning (DL) algorithms for position estimation.

The rest of the paper is structured as follows: Section II presents the comprehensive model pipeline of our WiFi-based indoor positioning system. Section III details the data collection and the processing techniques for both CSI and RSSI data. Section IV describes the application of Tabular Generative Adversarial Network (TabGAN or GAN) for data augmentation and highlights the implementation of DL algorithms, specifically Convolutional Neural Network (CNN) and LightGBM, on the augmented datasets. A thorough comparative analysis of the obtained results is conducted in Section V to assess the performance of each algorithm. Section VI concludes the paper.

II. Methodology

The proposed model pipeline consists of two essential phases: the offline phase and the online phase, as depicted in Figure 1.

In the offline phase, the focus is on training and data augmentation using a deep learning model. The initial step involves collecting real-world data by physically walking around the indoor environment and recording the RSSI at various locations. This dataset serves as the foundation for training the positioning algorithm. To enhance the training dataset, a Tabular GAN is employed for data augmentation. The Tabular GAN utilizes a generator network to learn the underlying distribution of the real data and generate synthetic data points resembling the collected RSSI and CSI measurements. The generated synthetic data, combined with the real data, forms an expanded and more diverse training dataset.

Moving to the online phase, when a user is in motion within the indoor environment, the system follows a series of steps for real-time positioning. First, the access points or beacons emit signals that are detected by the user's device, which measures the RSSI/CSI values. These measured values are then utilized in conjunction with the trained positioning algorithm. The algorithm compares the received RSSI and CSI values with the augmented training dataset, allowing for the estimation of the user's current position in real-time. By leveraging the augmented dataset and the positioning algorithm, which are in our case CNN and LightGBM, accurate and reliable indoor positioning can be achieved.

The combination of the offline phase, featuring deep learning-based data augmentation using Tabular GAN, and the online phase for real-time positioning facilitates dynamic and accurate indoor localization. This pipeline holds potential for a wide range of applications, including indoor navigation, asset tracking, and location-based services, offering improved accuracy and robustness in indoor positioning systems.

III. Dataset Creation

In this section, we will discuss the process of collecting data for the indoor positioning system, including the definition of the Raspberry Pi setup, the definition of the floor plan, and pinning the reference points.

A. Definition of the floor plan

In this work, data collection took place in the ITZ building of the University of Passau, in Germany. This indoor environment, spanning an area of approximately 47 square meters, served as the designated area for data collection.

By focusing on a specific location within the university building, the data collection process aimed to capture the unique characteristics and signal propagation patterns present in this particular indoor setting. The selected area provided a controlled environment for gathering data and conducting experiments, ensuring consistency and reproducibility in the collected dataset.

Each RP served as a designated location for data collection with specific coordinates within the room. The positioning of these reference points followed a regular pattern, with a distance of 1 meter between adjacent points and 0.45 meters from the walls. This configuration ensured that the RPs covered the entire area of the room, capturing the signal variations and characteristics at different positions.

To gather comprehensive data and capture signal variations from different directions within each RP, measurements were collected systematically from four cardinal directions: North, South, West, and East. This approach allowed for a more thorough assessment of the signal strength and characteristics in each RP. At each RP, the data collection process involved moving around the point and measuring the signal strength from the four specified directions. By collecting measurements from multiple directions, the dataset encompassed a wider range of signal variations, taking into account potential obstacles, signal blockages, or signal reflections from different angles.

B. Nexmon Firmware

For WiFi chips, Nexmon is a framework for firmware modification that makes it possible to enable extra features and capabilities above and beyond what stock firmware generally supports. To explore new possibilities and create cutting-edge applications, it gives researchers and developers the freedom to access and control WiFi chips' low-level features [7].

In the context of collecting RSSI and CSI data for fingerprinting and indoor localization, Nexmon can be used to capture and analyze the wireless signals transmitted by WiFi devices. By modifying the firmware on compatible WiFi chips, Nexmon allows for the extraction of detailed information about the wireless channel, including RSSI and CSI values. Using this information, fingerprints that depict the distinctive qualities of the wireless signals at various points in an indoor area can be made. These radio maps can be used as the foundation for IPS that use fingerprinting to locate a target device based on the characteristics of the received signal.

C. Data recording

The data recording process for collecting CSI data for fingerprinting and indoor localization using Nexmon on Raspberry Pi 4 involved several steps. First, the setup and configuration included using Raspberry Pi 4 with Nexmon firmware. Nexmon was configured to capture CSI data on channel 36 with a 80 MHz bandwidth, specifically targeting the first core of the WiFi chip and the first spatial stream. Next, the measurement procedure was conducted at various positions

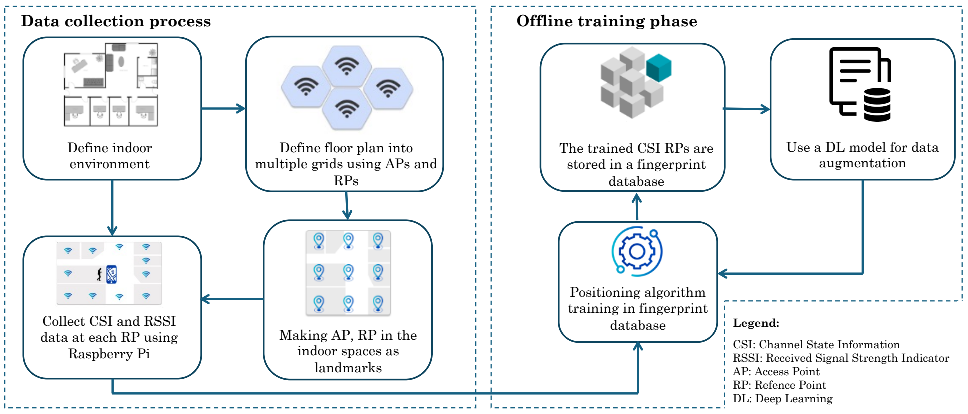
Data collection process
Define indoor environment
Define floor plan into multiple grids using APs and RPs
Collect CSI and RSSI data at each RP using Raspberry Pi
Making AP, RP in the indoor spaces as landmarks
Offline training phase
The trained CSI RPs are stored in a fingerprint database
Use a DL model for data augmentation
Positioning algorithm training in fingerprint database
Legend:
CSI: Channel State Information
RSSI: Received Signal Strength Indicator
AP: Access Point
RP: Refence Point
DL: Deep Learning

Fig. 1: Pipeline of the proposed model

within the target environment. For each position, 100 samples were collected to ensure accuracy and reliability. The Raspberry Pi 4 with Nexmon was carefully placed at each position, maintaining a stable and consistent setup throughout the data collection process. Measurements were taken in multiple directions (North, East, West, and South) to capture a comprehensive view of the wireless signals present. During the data collection phase, Nexmon on Raspberry Pi 4 listened on socket 5500 for User Datagram Protocol (UDP) packets, which contained the captured CSI data. The CSI data was extracted from these UDP packets, providing the necessary information for further analysis. The collected CSI data can be analyzed by opening the PCAP file in Wireshark or parsing it using a script.

1) Fingerprinting dataset: During the data collection phase, data capture was conducted within the ITZ building. The Raspberry Pi, fixed at a fixed height, was placed at various RPs to collect data in four directions. A total of 100 WiFi frames were captured for each reference point in each direction, resulting in 400 WiFi frames in total for all four directions. Since there were 45 reference points, the entire data collection process yielded approximately 18000 WiFi data frames.

The data collection process required two days to set up the RPs around the room and perform the measurements. During this time, two students collaborated to ensure the accurate positioning of the Raspberry Pi in each direction for every RP. The results of these measurements were stored in PCAP files for further analysis and processing.

In the dataset, we have a total of 28 unique MAC addresses. However, it is worth noting that five of these MAC addresses contribute significantly to the data, making up approximately 16,000 rows out of the total 18,000 rows in the dataset. These top five MAC addresses are responsible for a significant portion of the data and play a crucial role in the analysis. We have assigned a unique number to each MAC address, ranging from 1 to 28. This numbering scheme was implemented to facilitate the representation of MAC addresses in the bar chart.

D. Data preprocessing

To extract the CSI data from the PCAP files, the following preprocessing steps were performed using the provided code:

1) *Reading the PCAP file*: This step involved reading the PCAP file containing the captured wireless packets. This step extracted the necessary data from the PCAP file.
2) *Infer Bandwidth*: The bandwidth of the wireless signal was inferred from the length of the packets in the PCAP file. This ensured that the correct bandwidth was used for further processing.
3) *Determine Number of Subcarriers*: The number of Orthogonal Frequency-Division Multiplexing (OFDM) subcarriers was determined based on the inferred bandwidth. This value is required for correctly interpreting the CSI data.
4) *Estimate Maximum Number of Samples*: An estimate for the maximum possible number of samples in the PCAP file was calculated. This estimation is useful for allocating memory for storing the extracted data.
5) *Data Extraction*: The actual extraction of CSI data was performed by iterating over the packets in the PCAP file. For each packet, the relevant information such as RSSI, MAC ID, sequence number, core and spatial stream, and CSI data were extracted and stored.
6) *Conversion to Numpy Arrays*: The extracted CSI data and other relevant information were then converted to NumPy arrays for efficient processing and analysis. This conversion facilitated further manipulation and analysis of the data.
7) *SampleSet Object Creation*: Finally, all the extracted data were encapsulated in a data structure for easy access and analysis. This data structure provides convenient methods to retrieve specific information for a given sample index.

IV. Data Augmentation

In this section, we delve into the key aspect of our research, which is data augmentation using the Tabular GAN model.

A. Tabular Generative Adversarial Network (TabGAN) for Data Augmentation

Tabular GAN is a generative model specifically designed for augmenting tabular data. It leverages the power of GANs to generate synthetic data that closely resembles the original dataset, thereby increasing its size and diversity [8]. The architecture of the Tabular GAN comprises the following components: generator and discriminator.

1) Generator: is responsible for generating synthetic data that captures the distribution and characteristics of the original dataset. Its architecture consists of the following steps:

i. Generating Numerical Variables:
   - Scalar Value Generation: The Generator generates scalar values, such as those representing clusters, using techniques like sampling from a Gaussian Mixture Model (GMM).
   - Cluster Vector Generation: Cluster vectors are generated to represent the probability of each data point belonging to different clusters. These vectors can be obtained from GMM outputs or other methods.
   - Activation Function: Numerical variables are transformed using an activation function, such as the hyperbolic tangent (tanh), to ensure the generated values fall within a desired range.

ii. Generating Categorical Variables:
   - Probability Distribution: Categorical variables, such as labels or classes, are generated as probability distributions over all possible categories. Techniques like softmax activation function are used for this purpose.

iii. Long Short-Term Memory (LSTM) networks:
   - To generate rows effectively, an LSTM network with an attention mechanism is employed. This architecture enables the model to capture temporal dependencies and generate coherent synthetic samples.
   - Inputs to the LSTM include random variables, weighted context vectors, previous hidden states, and embedding vectors.

2) Discriminator: plays a vital role in distinguishing between real and synthetic data. It aims to learn the underlying patterns and characteristics of the original dataset. The Discriminator architecture consists of the following components:

i. Multi-Layer Perceptron (MLP):
   - The Discriminator utilizes an MLP with activation functions like LeakyReLU and techniques such as Batch Normalization. This MLP is responsible for extracting features and discriminating between real and synthetic data.
   - Concatenation: The numerical variables, cluster vectors, and binary variables obtained from the Generator are concatenated to form the input for the Discriminator.

ii. Loss Function:
   - The Discriminator's loss function incorporates components like the Kullback-Leibler (KL) divergence term and the sum ordinal log loss. These components allow the Discriminator to optimize its ability to differentiate between real and synthetic data effectively.

B. Evaluation of TabGAN

To assess the effectiveness of the augmented data generated by the Tabular GAN on the prediction of the position coordinates X and Y, we evaluate the performance using two different models: CNN and LightGBM.

We start by evaluating the performance of the CNN model on the following datasets:

1) *Ground Truth Data*: We assess the performance of CNN model on the original (ground truth) data. This serves as a baseline to compare against the performance on the augmented data. We calculate the Root Mean Square Error (RMSE) between the true values of the X and Y coordinates and the corresponding predictions made by the CNN model.
2) *Augmented Data*: Next, we evaluate the performance of CNN model when trained on the augmented data generated by the Tabular GAN. We use the same evaluation metric to compare the predictions made on the augmented data with the ground truth values. This step allows us to determine how well the Tabular GAN has captured the underlying distribution of the original data and whether the augmented data is useful for improving the prediction accuracy.
3) *Combined Data*: Finally, we assess the performance of CNN model when trained on a combination of the ground truth data and the augmented data. This step aims to investigate the potential benefits of incorporating the augmented data into the training process. We compute the RMSE between the predictions made on the combined dataset and the true values of the X and Y coordinates.

The evaluation process of LightGBM model is similar to the steps described above for CNN model.

C. CNN Architecture

In Figure 2, we present the detailed architecture of our CNN model. The CNN is designed to handle the positioning estimation task efficiently by processing the input data, extracting relevant features, and making accurate predictions.

The architecture comprises several essential components, including convolutional layers, pooling layers, and fully connected layers. These layers work collaboratively to learn hierarchical representations from the input data, enabling the model to capture intricate spatial patterns and relationships present in the RSSI and CSI measurements.

The initial convolutional layers act as feature extractors, convolving over the input data to detect spatial patterns and edges. The pooling layers then downsample the extracted features, reducing the computational complexity and aiding in learning spatial invariance.

Subsequently, the flattened feature maps are passed through fully connected layers, which serve as the decision-making units of the model. These layers combine the learned features

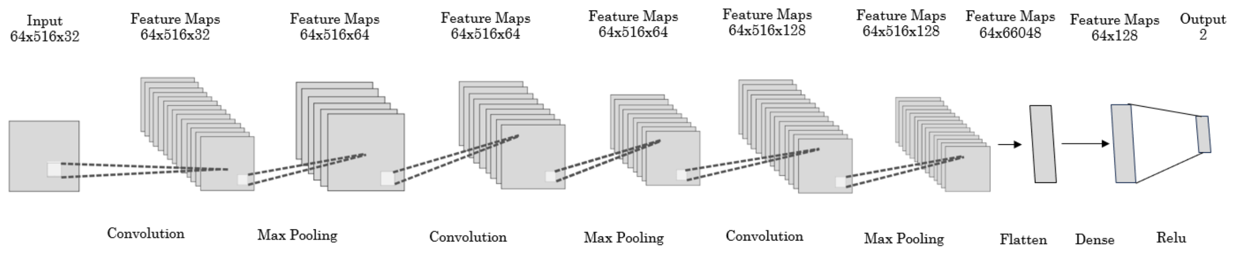

Fig. 2: CNN architecture

and apply non-linear transformations to make accurate positioning predictions.

Additionally, we have employed regularization techniques, such as dropout and batch normalization, to prevent overfitting and enhance model generalization. The CNN model is trained using an appropriate optimization algorithm, whis is Adam in our case, and a suitable loss function for regression tasks, such as RMSE.

The CNN architecture is designed to capture relevant spatial information and learn meaningful representations from the input data. By stacking convolutional and pooling layers, the model can hierarchically extract features and make predictions based on the learned patterns.

V. Results and Discussions

In this section, we present the evaluation and discussion of the system based on the Tabular GAN for data augmentation.

A. Data augmentation results

In order to validate the effectiveness of the data augmentation process using GANs, we conducted a comprehensive comparison of the data distribution. Specifically, we randomly selected a column from the original dataset and generated synthetic data using the GAN.

We noticed a striking resemblance between the data distribution of the original dataset and the data distribution of the generated synthetic data, demonstrates the GAN's ability to accurately capture and reproduce the underlying characteristics of the original data. This successful alignment further reinforces the efficacy of the data augmentation technique in preserving data distribution, making the generated data a valuable and reliable resource for enhancing our positioning estimation algorithm.

Through this comparison, we can confidently assert that the data augmentation process using GANs has effectively maintained the integrity of the original data's distribution. Such congruence is essential in ensuring that the generated data contributes meaningfully to the generalization and robustness of our positioning estimation model.

B. LightGBM for GAN evaluation

LightGBM, renowned for its efficiency and scalability, is particularly well-suited for tabular data analysis, making it an ideal choice for our positioning estimation problem [9].

First thing, our data was split as follows : 80% for train and 20% for test. After training LightGBM on the augmented dataset, which included the GAN-generated samples, we conducted an extensive evaluation using various regression-specific metrics, such as RMSE and Mean Positioning Error (MPE). RMSE allowed us to measure the average deviation between the predicted positioning coordinates and the ground truth values, providing a comprehensive assessment of the model's accuracy in estimating positions.

In our evaluation, we compared the performance of the LightGBM model on three different types of data: RSSI only, CSI only, and RSSI combined with CSI. We have used both CSI amplitude and phase. The results (see Table I) revealed intriguing insights into the significance of each data type for position estimation.

When training the model on the RSSI only data, we obtained an RMSE of 0.99 for X-coordinate, 2.6 for Y-coordinate and a MPE of 1.58 meters. Incorporating the GAN-generated samples through data augmentation improved the performance to an RMSE of 0.95 for X-coordinate and a MPE of 1.5 meters. Subsequently, when combining both RSSI and CSI data, the RMSE reduced to 0.914 for X-coordinate, 2.433 for Y-coordinate and the MPE to 1.5 meters.

C. CNN for GAN evaluation

Based on our feature importance analysis using LightGBM, we found that the combined RSSI and CSI data resulted in a slightly improved performance compared to using CSI data only. This observation highlights the importance of leveraging both RSSI and CSI features for accurate position estimation.

Given the insights from the feature importance analysis, we proceeded with the evaluation of the CNN model on the ground truth data and the augmented data. The CNN model was trained on the ground truth data to establish a baseline performance and assess its inherent capabilities in positioning estimation. Subsequently, we trained the CNN model on the augmented dataset, which included the GAN-generated samples, to evaluate the performance improvements brought about by GAN-based data augmentation.

In the process of training the model solely on the RSSI data, the resulting RMSE values were 0.76 for the X-coordinate and 2.01 for the Y-coordinate, which led to a MPE of 1.45 meters. Upon incorporating GAN-generated samples through data augmentation, the model's performance improved, resulting in an

TABLE I: LightGBM Model Evaluation

	Ground Truth Data		**Augmented Data**		**Combined Data**	
	RMSE (X/Y)	**MPE**	**RMSE (X/Y)**	**MPE**	**RMSE (X/Y)**	**MPE**
RSSI Only	0.99/2.6	1.58 m	0.95/1.555	1.5 m	0.914/2.433	1.5 m
CSI Only	0.795/1.289	1.242 m	0.6888/1.177	1.082 m	0.6487/1.0417	0.99 m
RSSI + CSI	0.796/1.289	1.242 m	0.6888/1.177	1.08 2m	0.6487/1.042	0.99 m

Legend: RMSE - Root Mean Square Error, MPE - Mean Positioning Error, m - meter

TABLE II: CNN Model Evaluation

	Ground Truth Data		**Augmented Data**		**Combined Data**	
	RMSE (X/Y)	**MPE**	**RMSE (X/Y)**	**MPE**	**RMSE (X/Y)**	**MPE**
RSSI Only	0.76/2.01	1.45 m	0.73/1.325	1.27 m	0.69/1,3	1.224 m
CSI Only	0.555/1.102	1.052 m	0.531/0.992	1.082 m	0.504/0.952	0.99 m
RSSI + CSI	0.554/1.1	1.04 m	0.529/0.971	0.97 m	0.507/0.93	0.8 m

Legend: RMSE - Root Mean Square Error, MPE - Mean Positioning Error, m - meter

RMSE of 0.73 for the X-coordinate and a MPE of 1.27 meters. Subsequently, when both RSSI and CSI data were combined, the RMSE reduced to 0.69 for the X-coordinate and 1.3 for the Y-coordinate, with a MPE of 1.224 meters. As seen in Table II, it is evident that the combination of both RSSI and CSI data yielded the most precise localization performance with a MPE of 0.8 meters.

The remarkable reduction in MPE for the augmented data further reinforces the superiority of GAN-based data augmentation in enhancing the model's performance. The significantly lower MPE demonstrates that the CNN, when trained on the augmented data, is better able to estimate the target positions with higher accuracy and precision, making it a compelling choice for positioning estimation tasks in real-world scenarios. The improved learning and generalization capabilities achieved with the augmented data reinforce the efficacy of GAN-based data augmentation in enhancing the CNN's performance for positioning estimation.

VI. Conclusions

In this paper, we embarked on a comprehensive investigation of an indoor localization system, leveraging WiFi data for precise positioning estimation. The data collection process was meticulously executed within a controlled lab environment, utilizing Raspberry Pi and Nexmon firmware to capture WiFi signals. Subsequently, we performed thorough data preprocessing to ensure data quality and consistency.

A key highlight of this work was the application of data augmentation using GAN to enrich the original dataset. The GAN-based augmentation technique effectively generated synthetic data points, enhancing the diversity and volume of the training data, which proved instrumental in improving the accuracy and generalization of our positioning algorithms.

Our evaluation process involved the utilization of two prominent DL algorithms: CNN and LightGBM. The results highlighted the remarkable improvements achieved when training on augmented data, demonstrating the efficacy of GAN-based data augmentation in boosting the precision of the positioning estimation. Our system showed the improvement of at least **25 cm** in positioning error when using GAN. Nonetheless, CSI combined with RSSI has shown a better influence with a positioning error equal to **0.8 meters**.

Future work in this domain could focus on expanding the evaluation to other machine learning algorithms and exploring different GAN architectures for data augmentation.

References

[1] W. F. Hassen and J. Mezghani, "Cnn based approach for indoor positioning services using rssi fingerprinting technique," in *2022 International Wireless Communications and Mobile Computing (IWCMC)*. IEEE, 2022, pp. 778–783.

[2] W. F. Hassen and F. Najjar, "A positioning handoff decision algorithm for ubiquitous pedestrian navigation systems," pp. 487–494, 2016.

[3] W. F. Hassen, L. Brunie, A. Nechba, and H. Kosch, "Continuous indoor/outdoor pathway display algorithm for pedestrian navigation service," in *Proceedings of the 17th International Conference on Advances in Mobile Computing & Multimedia*, 2019, pp. 66–73.

[4] W. F. Hassen, F. Najjar, L. Brunie, H. Kosch, and Y. Slimani, "Smart pdr integration for ubiquitous pedestrian navigation service," in *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*. IEEE, 2017, pp. 1558–1563.

[5] P. Roy and C. Chowdhury, "A survey of machine learning techniques for indoor localization and navigation systems," *Journal of Intelligent & Robotic Systems*, vol. 101, p. 63, 2021.

[6] A. Poulose and D. S. Han, "Hybrid deep learning model based indoor positioning using wi-fi rssi heat maps for autonomous applications," *Electronics*, vol. 10, no. 1, 2021. [Online]. Available: https://www.mdpi.com/2079-9292/10/1/2

[7] Seemoo Lab, "NEXMON: The C-based Firmware Patching Framework," https://github.com/seemoo-lab/nexmon, 2023.

[8] P. F. Moshiri, H. Navidan, R. Shahbazian, S. A. Ghorashi, and D. Windridge, "Using gan to enhance the accuracy of indoor human activity recognition," *arXiv preprint arXiv:2004.11228*, 2020.

[9] L. Yin, P. Ma, and Z. Deng, "Jlgbmloc—a novel high-precision indoor localization method based on lightgbm," *Sensors*, vol. 21, no. 8, p. 2722, 2021.