

SURIMI: Supervised Radio Map Augmentation with Deep Learning and a Generative Adversarial Network for Fingerprint-based Indoor Positioning

Darwin Quezada-Gaibor^{*,†}, Joaquín Torres-Sospedra[‡],
Jari Nurmi[†], Yevgeni Koucheryavy[†], and Joaquín Huerta^{*}
^{*}*Institute of New Imaging Technologies, Universitat Jaume I, Castellón, Spain*
[†]*Electrical Engineering Unit, Tampere University, Tampere, Finland*
[‡]*Algoritmi Research Centre, Universidade do Minho, Guimarães, Portugal*

Abstract—Indoor Positioning based on Machine Learning has drawn increasing attention both in the academy and the industry as meaningful information from the reference data can be extracted. Many researchers are using supervised, semi-supervised, and unsupervised Machine Learning models to reduce the positioning error and offer reliable solutions to the end-users. In this article, we propose a new architecture by combining Convolutional Neural Network (CNN), Long short-term memory (LSTM) and Generative Adversarial Network (GAN) in order to increase the training data and thus improve the position accuracy. The proposed combination of supervised and unsupervised models was tested in 17 public datasets, providing an extensive analysis of its performance. As a result, the positioning error has been reduced in more than 70% of them.

Index Terms—generative networks, indoor positioning, machine learning, Wi-Fi fingerprinting

I. INTRODUCTION

Since the early years of the 21st century special emphasis is placed on the design of Indoor Positioning Systems (IPSs) based on Machine Learning (ML) models and Wi-Fi fingerprinting [1]–[3] which include supervised, semi-supervised, unsupervised and reinforcement learning models [4], [5]. Although Wi-Fi fingerprinting is very popular, it suffers from scalability problems in terms of position accuracy and time response [6].

In general, Wi-Fi Fingerprinting is divided into two well-defined phases: the offline phase where the radio map is generated by collecting fingerprints at known reference points, and the online phase where the incoming fingerprint (at an unknown position) is compared to the fingerprints in the radio map in order to estimate the device position [7]. Therefore, in the traditional fingerprinting approach [8], the computational cost of the latter one is highly dependent on the radio map size.

Corresponding Author: D. Quezada Gaibor (quezada@uji.com)

The authors gratefully acknowledge funding from European Union's Horizon 2020 Research and Innovation programme under the Marie Skłodowska Curie grant agreements No. 813278 (A-WEAR: A network for dynamic wearable applications with privacy constraints, <http://www.a-wear.eu/>) and No. 101023072 (ORIENTATE: Low-cost Reliable Indoor Positioning in Smart Factories, <http://orientate.dsi.uminho.pt>).

Some researchers have proposed alternatives balancing between the positioning accuracy and execution time and, thus, diminishing the scalability problems in Wi-Fi fingerprinting. For instance, Song *et al.* [9] proposed a new solution for Wi-Fi fingerprinting based on the combination of Stacked Auto-Encoder (SAE) and CNN, to improve the accuracy of the building and floor detection, and therefore, the position accuracy. Torres-Sospedra *et al.* [10] provided an analysis of clustering algorithms in Wi-Fi fingerprinting –in terms of positioning accuracy and computational costs– which included an evaluation framework with many open-source radio maps.

It is well-known that creating an accurate radio-map is time-consuming and requires extensive manual-labour. However, the radio map should be updated after any significant change in the environment to guarantee the position accuracy [11]. An alternative to reduce the time for a radio map collection is to generate *artificial* fingerprints using mathematical or statistical models [12]. Currently, some ML models (e.g., GANs) are capable of generating new data, including artificial Received Signal Strength (RSS) values for radio map augmentation [13], which cannot be easy to identify whether they are real or not.

In this article we propose the combination of deep learning models (CNN and LSTM) and conditional GAN, in order to enhance the radio map without additional manual labour, and therefore, to reduce the positioning and localization error. The main contributions of this work are the following:

- A new generalized machine learning model for data augmentation and indoor positioning (CNN-LSTM-cGAN).
- An extended analysis of the proposed model with several Wi-Fi and Bluetooth Low Energy (BLE) datasets.
- The source-code of the CNN-LSTM-cGAN model [14].

The remaining of this paper is organized as follows. Section II provides a general overview of indoor positioning based on deep learning models. Section III describes the proposed deep learning model for Wi-Fi fingerprinting. Section IV provides the details of experiments carried out and their results. Section V offers a general discussion of the results. Finally, section VI provides the main conclusions of this research article.

II. RELATED WORK

Wi-Fi fingerprinting technique has been widely used for indoor positioning purpose due to its low-cost and also the support of modern user devices such as smartphones, wearable and Internet of Things (IoT) devices to Wi-Fi technology [15], [16]. However, Wi-Fi fingerprinting is not as accurate as ultra-wideband (UWB) or Visible light communication (VLC), which can reach centimetre level accuracy [17], [18]. With the aim of reducing the positioning error and enhance the scalability of Wi-Fi fingerprinting some authors have proposed multiple solutions that contemplate the use of deep learning models such as recurrent neural networks (RNN), CNN, LSTM, among others [19], [20].

Deep learning models are specially used for feature learning, allowing the extraction of meaningful information from the raw data. In this case, CNN, LSTM or other models are used to extract the main information of the radio map and then estimate the user's localization. For instance, Song *et al.* [21] combined SAE and CNN to provide a highly accurate indoor localisation solution which achieved 100% of the building rate and 95% of the floor rate approximately. The authors used the SAE to extract key characteristics of the RSS values in the dataset, and the CNN model for high accurate classification.

Recent models such as GANs [22] are not only used for generating new images or text but also, these models are used to generate new RSS values on the basis of real data. Belmonte-Hernández *et al.* [23] offered a novel framework which incorporates the advantages of the RNN and GAN in order to recover the information when the measurements are not received from the devices deployed in the environment. The recovered information is used to compute the user path and update the radio map. As a result, the positioning error was reduced by 3% in comparison to other techniques.

Li *et al.* [24] proposed a new model namely Amplitude-Feature Deep Convolutional GAN (AF-DCGAN). In this research work, the authors used the Channel State Information (CSI) to enhance the radio map, and transformed each amplitude feature map into an image for further processing. Thereby, the proposed model was used to generate new images based on the original image and these new amplitude feature maps are added to the original one to reduce the positioning error.

Njima *et al.* [13] used deep learning models and GAN to generate new RSS data in order to extend the radio map. Unlike the earlier research, work published in [13] has established a new criteria to select the most realistic fake fingerprints also known as artificial or synthetic fingerprints. As a result, the authors have shown an improvement in localization accuracy of 15% in comparison with the benchmark.

This paper offers a new framework that includes the positioning and GAN models to generate realistic fingerprints and improve the training stage in the offline phase of Wi-Fi fingerprinting. Unlike the previous research work, the GAN model consists of a conditional label (floor label) used as an extra parameter in the generative model, allowing to generate artificial data for multi-building and multi-floor environments.

III. DATA AUGMENTATION FRAMEWORK

This section provides a general description of the proposed framework and their components.

A. Indoor Positioning Framework description

The proposed indoor positioning framework based on Wi-Fi fingerprinting is devoted to increase the fingerprints in the radio map and reduce the positioning error using three well-known machine learning models CNN, LSTM and cGAN. The CNN-LSTM model is used to predict the 2D/3D position, including floor and building, while the cGAN is used to generate *artificial* fingerprints to enrich the radio map. These ML models use only the information stored in the radio map, which include the RSS values, X, Y and Z position, and floor & building tags. Once the artificial fingerprints are generated the position for each fingerprint is predicted and only the most relevant fingerprints are added to the original dataset to form an enriched radio map (see Fig. 1).

B. Fingerprinting

Indoor positioning based on Wi-Fi fingerprinting consists of two main phases as mentioned in earlier paragraphs. In the offline stage, the radio map is formed and the proposed machine learning model is trained in order to be used in the next stage. In the online phase, the user position is predicted.

Generally, a radio map (Ψ) is formed by m number of samples – fingerprints – and n number of Access Points (APs) ($m \times n$). Each position in the radio map is corresponding to a RSS value (ψ_{ij}) in the i -th position or sample ($i = 1, 2, \dots, m$) and transmitted by the n -AP ($j = 1, 2, \dots, n$).

$$\Psi = \begin{bmatrix} \psi_{11}, \psi_{12} & \dots & \psi_{1n} \\ \psi_{21}, \psi_{22} & \dots & \psi_{2n} \\ \vdots & \ddots & \vdots \\ \psi_{m1} & \dots & \psi_{mn} \end{bmatrix}$$

Each sample (fingerprint) in the dataset is linked to a known training position (X, Y, Z, floor and building), these labels are used during the training stage.

C. Data preprocessing

Data preprocessing is fundamental in the early stages of data analysis and machine learning, using multiple techniques for data cleaning, scaling, encoding, among other. Therefore, before data processing we have changed the data representation of each dataset as is suggested in [10], [25]. The new data representation reduce the data complexity allowing to extract more useful information with ML models. Additionally, *MinMaxScaler* is used to scale the longitude, latitude and altitude labels – X, Y and Z – (see Eq. 1) and *OneHotEncoder* for floor and building.

$$X_{scaler} = \frac{X - X_{min}}{X_{max} - X_{min}} \quad (1)$$

where, X_{min} represents the minimum value and X_{max} is the maximum value in the labels of longitude, latitude and altitude.

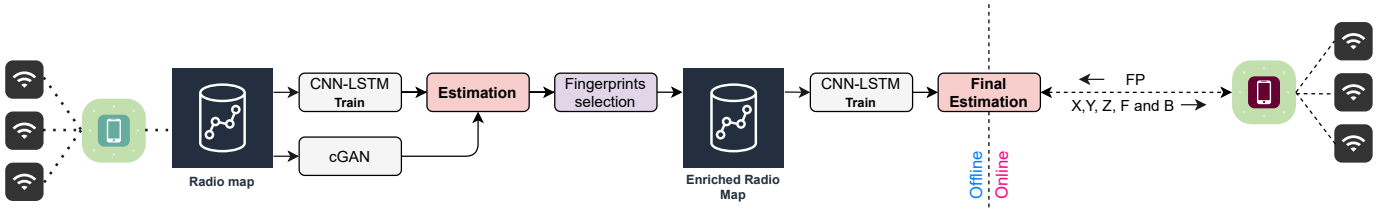


Fig. 1. CNN-LSTM-cGAN BLE/Wi-Fi fingerprinting process

D. Positioning model - CNN-LSTM

Given the heterogeneity of indoor environments, user devices and the fluctuation of the Wi-Fi signal over the time makes deep learning models useful to learn those patterns or changes [13]. The first layer of the proposed CNN-LSTM model is composed of a *Conv1D* layer that helps to extract spatial characteristics of the input data. Then, we have a *max pooling* layer (MaxPooling1D), which allows computing the maximum value for each patch in the feature map. To reduce overfitting problems, a *dropout* layer is added to the positioning model, the dropout rate is 0.5 for all the models. The dropout layer is used to deactivate some neurons during the training stage. The next three layers are a Conv1D, MaxPooling1D and dropout with similar characteristics to the previous layers. The following layer is the *flatten* layer that converts the data into a 1-dimensional array. The next layer is the *LSTM* used to learn order dependence. Finally, we have the *dense* layer fully connected which perform the prediction of the 2D/3D position (longitude, latitude and altitude).

The model to classify the fingerprints per floor is similar to the positioning model, but with different values for the filter, activation function, and learning rate, among others. In the case of the building model, it is less complex in comparison with the previous models and it uses only one convolutional layer. Table I shows the positioning model layers with their parameters and values. Additionally, we adopt early stopping (*patience* = 5) to avoid overfitting.

E. Data augmentation

Data augmentation is done by using the cGAN architecture, which contains two main components (models), the discriminator and generator. Unlike GAN, cGAN includes a conditional label (class) allowing the targeted generation of fingerprints of a given floor or building. The primary function of the discriminator is to distinguish or classify the fingerprints as either artificial or real (i.e., it is a binary classifier of either 0 or 1). The generator is devoted to generating new fingerprints similar to real fingerprints (see Fig. 2).

In this architecture, both models (discriminator and generator) are trained together in an adversarial manner. It means that the discriminator capabilities increase at the expense of decreasing the capabilities of the generator and vice-versa also known as min-max game (see Eq. 2). As a result, new original fingerprints are generated in locations close to real ones.

TABLE I
CNN-LSTM PARAMETER

Layer	Parameter	Position	Floor	Building	
TimeDistributed	Conv1D	Filter	8	16	16
		Activation	elu	relu	relu
		kernel_size	1	1	1
	MaxPooling1D	pool_size	1	2	2
	Dropout	rate	0.5	0.5	0.5
	TimeDistributed	Conv1D	Filter	8	32
Activation			elu	relu	
kernel_size			1	1	
padding			same	same	
MaxPooling1D	pool_size	1	1		
Dropout	rate	0.5	0.5		
Flatten					
LSTM	Units	40	50	40	
	Activation	elu	relu	relu	
Dense	Units	3	CF	CB	
	Activation	elu	softmax	softmax	
Training parameters	lr	0.0005	0.0001	0.0001	
	epochs	100	100	100	
	batch_size	256	100	100	
	Optimizer	Adam	Adam	Adam	

CF: Number of classes - Floors, CB: Number of classes - Buildings

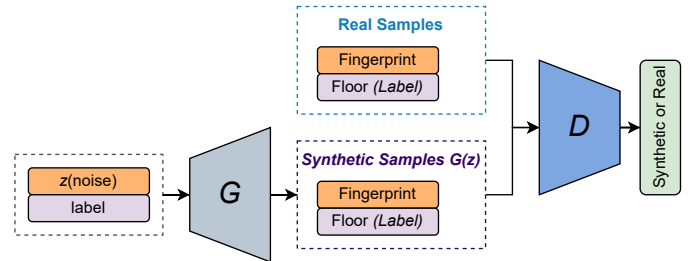


Fig. 2. cGAN architecture

$$\min_G \max_D V(D, G) = \mathbb{E}_{\Psi \sim p_{data}(\Psi)} [\log D(\Psi|y)] + \mathbb{E}_{z \sim p_z(z)} [\log 1 - D(G(\Psi|y))] \quad (2)$$

where, $V(D, G)$ is the value function, D represents the discriminator and G is the generator. x represents the fingerprints, y the classes, z the noise values. $p_{data}(x)$ is the data distribution over the data x and $p_z(z)$ is the noise distribution.

Fig. 3 shows the layers of the discriminator and generator model together with some of their parameters. These ML models contain convolutional layers (Conv1D), dense layers and flatten layers. In the same way as the positioning model, dropout layers were added to each model in order to reduce the overfitting. Unlike the discriminator, the generator model uses Conv1DTranspose instead of Conv1D, which is the inverse operation of Conv1D. Additionally, both the discriminator and generator use Leaky Rectified Linear Unit (LeakyReLU) activation function. LeakyReLU is mathematically defined as (see Eq. 3) [26].

$$\mu_i = \begin{cases} x_i, & \text{if } x_i \geq 0, \\ \frac{x_i}{a_i}, & \text{if } x_i < 0 \end{cases} \quad (3)$$

where, μ_i represents the output after passing the LeakyReLU function, a_i is a fixed parameter between 1 to ∞ and x_i denotes the input data.

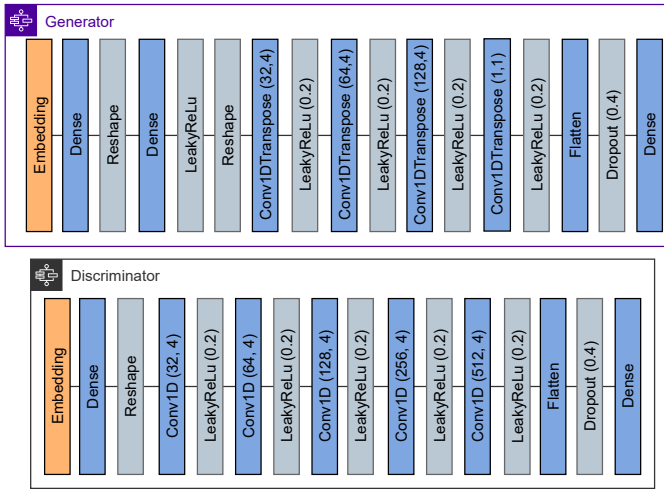


Fig. 3. Discriminator and generator model

The following parameters were used to compile both the discriminator and generator model: Adam optimizer, learning rate equal to 0.0002 and binary cross-entropy loss.

F. Fingerprints selection

Given that the position of the new fingerprints is unknown and can not be parameterized during the training stage, a new algorithm is designed to select the most relevant synthetic fingerprints. The cGAN network thus generates n number of new fingerprints and then the proposed algorithm selects only those fingerprints close to the real ones (see Algorithm 1). Once the fingerprints are generated and selected, the CNN-LSTM model is trained with the augmented radio map to increase the prediction and classification accuracy.

The proposed algorithm requires five input parameters: the training fingerprints (Ψ_{XTR}) and their labels (Ψ_{yTR}), the number of artificial fingerprints to be generated (\aleph_{nfs}), and a list of distances (\aleph_d) which is used to select the most relevant fingerprints after n number of iterations (\aleph_i). For each iteration, the algorithm generates a latent space (i.e., the

latent space is a Gaussian distribution with zero mean and standard deviation of one). In this case, the latent space is a $\aleph_{nfs} \times n$ matrix (l_p) (n is equal to the number of features in the radio map) and their corresponding labels (l_i). The latent space is used to generate the synthetic fingerprints (Ψ'_{XF}). Once the new fingerprints are generated, the CNN-LSTM is used to predict their positions (X, Y, Z , floor and building) (Ψ'_{yF}). In order to select the most relevant fingerprints, the distance between the positions of artificial and real fingerprints (Ψ'_{yF} and Ψ'_{yTR} respectively) is computed in order to create a distance matrix ($D_{ij} \in \mathcal{R}^{\aleph_{nfs} \times m}$, where m is the number of samples in the training dataset). This distance matrix help us to determine whether there is at least one artificial fingerprint close to a real one ($\Psi_{XF_i} \leftarrow \Psi'_{XF_i} \Leftrightarrow \exists d_{ij}, d_{ij} \leq dist$, where d_{ij} is the computed distance in the i -th and j -th position in the distance matrix and $dist$ is the distance used to select the artificial fingerprints). Thus, only those fingerprints in the established range will be selected to enrich the radio map.

The output parameters are the artificial fingerprints (Ψ_{XF}) and their corresponding labels (Ψ_{yF}).

Algorithm 1: Fingerprints selection

Input: $\Psi_{XTR}, \Psi_{yTR}, \aleph_d, \aleph_{nfs}, \aleph_i$
Output: Ψ_{FX}, Ψ_{Fy}

```

1 Function artificialFPSelection( $\Psi_{XTR},$ 
   $\Psi_{yTR}, \aleph_d, \aleph_i, \aleph_{nfs}$ ):
2   for  $dist$  in  $\aleph_d$  do
3     for  $iter = 0$  to  $\aleph_i$  do
4       /* Latent space  $l_p$  */
5        $l_p \in \mathcal{R}^{\aleph_{nfs} \times n}$ 
6       /* Labels */
7        $l_i \in \mathcal{R}^{\aleph_{nfs}}$ 
8        $\Psi'_{XF} = \text{cgan.predict}([l_p, l_i])$ 
9       /* Predict position ( $X, Y, Z$ ),
10      floor and building ( $\Psi'_{yF}$ ) */
11       $\Psi'_{yF} = \text{cnn\_lstm.predict}(\Psi'_{XF})$ 
12      /* Distance matrix between the
13      real and artificial
14      fingerprints. */
15       $D_{ij} = D(\Psi'_{yF}, \Psi'_{yTR}) = \|\Psi'_{yF} - \Psi'_{yTR}\|^2$ 
16       $D_{ij} \in \mathcal{R}^{\aleph_{nfs} \times m}$ 
17      /* Selecting relevant
18      artificial fingerprints */
19       $\Psi_{XF_i} \leftarrow \Psi'_{XF_i} \Leftrightarrow \exists d_{ij}, d_{ij} \leq dist$ 
20    end
21  end
22  return  $\Psi_{XF}, \Psi_{yF}$ 
23 End Function

```

G. Training Method Description

Three different methods have been used to train the cGAN network in order to identify the optimal parameters to generate new fingerprints.

1) *First Method (M1)*: This method consists of training the model per building using floor label as the conditional parameter in the cGAN network.

2) *Second Method (M2)*: In this method, the training stage is done per floor and the conditional parameter is the building label.

3) *Third Method (M3)*: This method uses the whole dataset to train the cGAN network, and the conditional label is the floor. Unlike of the previous methods, this third method does not take into account the building label during the cGAN training stage.

IV. EXPERIMENTS AND RESULTS

A. Experiment setup

The experiments were executed on a computer with Fedora Linux 32, Intel® Core™ i7-8700T processor and 16 GB of RAM. The methods were implemented in Python 3.9.

A total of fifteen public Wi-Fi and two BLE fingerprint datasets were used to test the proposed framework for indoor positioning and data augmentation: UJI 1–2 (UJIIndoorLoc), LIB 1–2, UJIB 1–2 (Universitat Jaume I); TUT 1–7 (Tampere University); DSI 1–2 (University of Minho); MAN 1–2 (University of Mannheim) and UTSIndoorLoc (UTS) (University of Technology Sydney) [10], [21], [27]. All datasets were independently collected, ensuring the experiments are based on heterogeneous datasets and the results can be generalised. Three datasets were used to fine-tune the proposed model (hyperparameters election) and the remaining ones were used to assess the general performance.

Since the data representation may influence the performance of machine learning models, we have selected *powred* data representation (see Eq. 4) to minimize the signal fluctuation existing in the datasets prior training the proposed positioning and data augmentation model.

$$Powred = \begin{cases} 0, & \text{if } RSS_i = 0, \\ \left(\frac{RSS_i - \min(\Psi)}{-\min(\Psi)} \right)^\beta, & \text{otherwise} \end{cases} \quad (4)$$

where i is the i -th RSS value, $\min(\Psi)$ is the lowest RSS value in the dataset and β is the mathematical constant e .

Once the data is normalized, the positioning model (CNN-LSTM) is trained to predict the user or device position. To avoid the overfitting and overtraining issues the *early stopping* method has been used during the training stage. Fig. 4 shows the training loss (blue) vs. the validation loss (orange) of the positioning model for the UJI 1 dataset.

The first estimator of the proposed framework (with the original radio map) was tested in terms of 2D and 3D mean positioning error (ϵ_{2D} and ϵ_{3D}) and floor hit rate (γ). It is compared to benchmark methods proposed in the literature by Song *et al.* [21] (see Table II).

The results in Table II show that the proposed CNN-LSTM provides a slightly reduction of the positioning error with respect to CNNLoc [9] in UJI 1 and UTS datasets, but the error increased in the TUT 3 dataset. The results of our CNN-LSTM estimator are promising even without data augmentation.

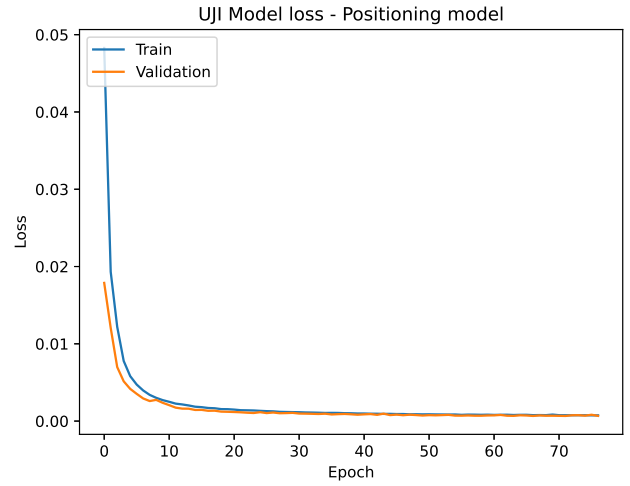


Fig. 4. CNN-LSTM model

TABLE II
BENCHMARK AND POSITIONING MODEL(CNN-LSTM)

Database	Dim.			CNNLoc [9]		CNN-LSTM	
	$ \tau $	$ A $	δ_{fp}	ϵ_{2D} [m]	ϵ_{3D} [m]	ϵ_{2D} [m]	ϵ_{3D} [m]
*UJI 1	20982	520	1–77	11.78	–	11.17	11.24
*TUT 3	3951	992	1–2	10.88	–	11.14	11.23
*UTS	9496	589	1–35	7.60	–	7.34	7.57

*Multi-building Multi-floor, •Single-building Multi-floor, $|\tau|$: number of fingerprints, $|A|$: number of APs, δ_{fp} : number of fingerprints per reference point, ϵ_{2D} : mean positioning error 2D, ϵ_{3D} : mean positioning error 3D.

B. Setting the hyperparameters for the GAN network

The proposed CNN-LSTM is working for position estimation using deep learning. However, we consider that its performance can be improved by augmenting the radio map. Thus, we have used three reference datasets found in the literature (UJI 1, TUT 3 and UTS) to select the optimal hyperparameters values to train the proposed cGAN model.

Table III shows the training parameters for the cGAN architecture and the results obtained with each method and configuration. The number of epochs for each test was set to 14, and the batch size to 64 and 128. The number of iterations is 10, and the distance between the real samples and the *artificial* samples is between 1 m to 5 m or 1 m to 10 m. It means that 200 samples were generated for each meter, and only those new fingerprints in the defined range were selected.

1) *Results – First Method*: M1 was tested only in the UJI 1 dataset due to that this dataset contains samples collected in 3 buildings and 4–5 floors per building. The first configuration used was the following: batch size equal to 64, and distance between 1 m to 5 m. As a result, the positioning error was reduced by 11% in comparison with the error reported by [21]. When the maximum distance between real and synthetic fingerprints is set to 10 m the error slightly increased by 6% approximately with regards to the previous configuration.

TABLE III
TRAINING PARAMETERS FOR THE CGAN AND RESULTS OBTAINED WITH EACH METHOD AND CONFIGURATION FOR THREE DATASETS

Database	Training Parameters					Positioning			
	Epoc.	BS	Method	Iter.	Dist.	η	ϵ_{2D}	ϵ_{3D}	
UJI 1	14	64	M1	10	1-5	155	10.01	10.07	
	14	64	M1	10	1-10	1415	10.64	10.70	
	14	64	M2	10	1-5	213	10.55	10.60	
	14	64	M2	10	1-10	1398	11.18	11.25	
	14	64	M3	10	1-5	288	10.85	10.91	
	14	64	M3	10	1-10	11 138	11.00	11.06	
	14	128	M1	10	1-5	311	10.67	10.73	
	14	128	M1	10	1-10	2088	11.80	11.84	
	14	128	M2	10	1-5	311	11.25	11.31	
	14	128	M2	10	1-10	2088	11.93	11.99	
	14	128	M3	10	1-5	132	10.97	11.02	
	14	128	M3	10	1-10	1677	11.60	11.66	
	TUT 3	14	64	M2	10	1-5	848	10.23	10.44
		14	64	M2	10	1-10	5175	9.62	9.72
14		64	M3	10	1-5	3951	9.67	9.76	
14		64	M3	10	1-10	2912	10.13	10.23	
14		128	M2	10	1-5	600	11.41	11.51	
14		128	M2	10	1-10	3498	9.26	9.34	
14		128	M3	10	1-5	449	11.50	11.61	
14		128	M3	10	1-10	4847	9.77	9.86	
UTS	14	64	M2	10	1-5	266	7.21	7.57	
	14	64	M3	10	1-5	389	7.48	7.68	

In the second configuration ($batchsize = 128$ and distance between 1 m to 5 m) the results obtained are less accurate than the first configuration increasing the mean positioning error by more than 6%. Similarly, the mean positioning error also increased (more than 1 m) when the maximum distance between real and synthetic fingerprints is 10 m.

Fig. 5 illustrates a real example with the position of the reference fingerprints collected empirically (blue dots) and the new fingerprints generated with the GAN (red dots) for the UJI 1 datasets. The cGAN was trained per each building (as defined in the first method) and the distance between the real and new fingerprints is less than 5 m.

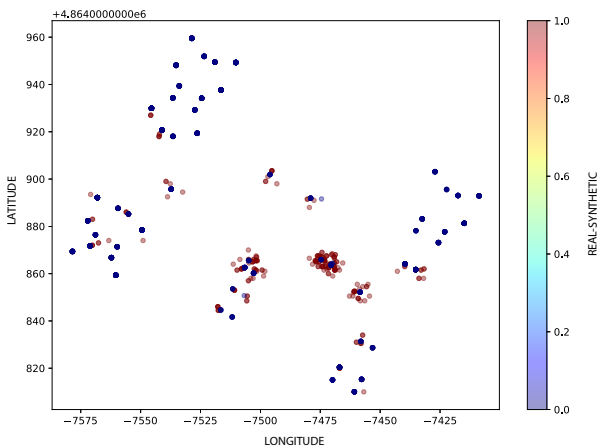


Fig. 5. Location of the reference fingerprints for the UJI 1 dataset after applying our generative model (Building 1 and Floor 0).

2) *Results – Second Method:* The mean positioning error obtained after training the GAN network by floor were less than the benchmark. However, it was higher than the first method in the case of the UJI 1 dataset (see Table III).

Similarly, the mean positioning error was reduced by 6% in TUT 3 dataset using 14 epochs, batch size equal to 64 and distance from the real samples is from 1 m to 5 m. If the distance between samples increased up to 10 m, the positioning error decreased by more than 11%. In the case of UTS, the positioning error was reduced by 5%.

3) *Results – Third Method:* The third method provides lower performance than the first and second methods. However, the positioning error is still lower than the benchmark. For instance, the positioning error was reduced by 8% approximately in the UJI 1 dataset.

In the case of TUT 3 dataset, 3951 new *artificial* fingerprints were created with the following training configuration: 14 epochs, batch size equal to 64 and distance from the real fingerprints between 1 m to 5 m; thus, the mean 2D position error was reduce by 11% (approx.), from 10.88 m to 9.67 m in comparison with the mean 2D positioning error reported by [9]. When the GAN is trained with 14 epochs, the batch size 128, and the distance less than 10 m, the positioning error increased by 10 cm despite the fact that the number of artificial fingerprints was more than the previous configuration.

Additionally, the cGAN network was trained using a batch size equal to 128, 14 epochs and distance was from 1 m to 5 m. If we compare the error obtained with the same parameters but the batch size equal to 64, the error increase a few centimetres (11 cm approx.) in both 2D and 3D positioning.

C. Generalized results

In Section IV-B multiple tests have been done in 3 reference Wi-Fi fingerprint datasets in order to get the cGAN training hyperparameters. In general, the three methods (M1–M3) have good performance using the batch size equal to 64 and distance between 1 m to 5 m. These parameters were tested in 14 additional public datasets in order to determine if the suggested configuration can be used in multiple datasets. Unlike the previous experiments, only M2 and M3 were used in the additional datasets, as none of them is multi-building.

To compare the proposed framework, we used the k -nearest neighbors (k -NN) algorithm as the baseline positioning method (see Table IV). We thus report the normalized values of the 2D and 3D mean positioning error ($\bar{\epsilon}_{2D}$ and $\bar{\epsilon}_{3D}$) to enhance the relative impact on each dataset [28].

1) *Results - M2:* After applying the second method the mean positioning error was reduced by more than 20% in DSI 1, LIB 2 and TUT 1 datasets compared with the k -NN. However, there are some cases where the positioning error increased with regards to the CNN-LSTM, but, it is not higher than 6%. Similarly, the 2D positioning error increased in TUT 2-7 datasets, but the 3D positioning was slightly reduced in TUT 2 dataset. Furthermore, the positioning model improved the floor hit rate by 3% on average (see Table IV).

TABLE IV
COMPARISON AMONG THE 1-NN BASELINE, CNN-LSTM AND THE PROPOSED MODELS. ϕ STANDS FOR THE NUMBER OF NEW FINGERPRINTS.

Database	Dim.			Baseline 1-NN					CNN-LSTM			SURIMI M2			SURIMI M3				
	$ \mathcal{T} $	$ \mathcal{A} $	δ_{fp}	ϵ_{2D} [m]	ϵ_{3D} [m]	$\bar{\epsilon}_{2D}$ [-]	$\bar{\epsilon}_{3D}$ [-]	γ [%]	$\bar{\epsilon}_{2D}$ [-]	$\bar{\epsilon}_{3D}$ [-]	γ [%]	ϕ	$\bar{\epsilon}_{2D}$ [-]	$\bar{\epsilon}_{3D}$ [-]	γ [%]	ϕ	$\bar{\epsilon}_{2D}$ [-]	$\bar{\epsilon}_{3D}$ [-]	γ [%]
Wi-Fi datasets																			
DSI 1	1369	157	6	4.97	4.97	1	1	100.00	0.88	0.88	100.00	25 712	0.74	0.74	100.00	24 358	0.79	0.79	100.00
DSI 2	576	157	2 to 3	4.96	4.96	1	1	100.00	1.29	1.29	100.00	24 350	0.98	0.98	100.00	23 897	1.02	1.02	100.00
LIB 1	576	174	12	3.00	3.01	1	1	99.84	0.96	1.02	99.36	3712	0.94	0.95	98.85	54	0.94	0.98	99.07
LIB 2	576	197	12	4.00	4.18	1	1	97.67	0.73	0.74	99.42	969	0.76	0.77	98.33	118	0.80	0.86	99.62
MAN 1	14 300	28	110	2.84	2.84	1	1	100.00	0.87	0.87	100.00	23 578	0.87	0.87	100.00	24 247	0.89	0.89	100.00
MAN 2	1300	28	10	2.47	2.47	1	1	100.00	0.98	0.98	100.00	11 478	0.84	0.84	100.00	6505	0.86	0.86	100.00
TUT 1	1476	309	1	8.59	9.55	1	1	90.00	0.88	0.81	93.27	4541	0.86	0.78	89.18	3526	0.83	0.76	91.63
TUT 2	584	354	1	13.00	15.11	1	1	71.02	1.05	0.90	90.34	613	1.03	0.89	89.77	45	1.14	0.98	90.91
TUT 4	3951	992	1	6.15	6.40	1	1	95.40	1.02	0.98	96.41	5502	1.03	1.00	96.84	184	1.16	1.13	96.70
TUT 5	446	489	1	6.39	6.92	1	1	88.39	1.74	1.62	97.56	596	1.74	1.62	98.57	1232	1.49	1.38	99.29
TUT 6	3116	652	1	2.07	2.08	1	1	99.95	2.62	2.62	99.90	6329	2.10	2.10	99.39	186	2.60	2.60	99.96
TUT 7	2787	801	1	2.23	2.62	1	1	99.12	2.32	1.98	98.06	809	2.18	1.86	98.00	82	2.17	1.86	98.32
BLE datasets																			
UJIB 1	1632	24	30 to 36	3.07	3.07	1	1	100.00	0.93	0.93	100.00	1990	0.95	0.95	100.00	1990	0.95	0.95	100.00
UJIB 2	816	22	24	4.25	4.25	1	1	100.00	0.93	0.93	100.00	1990	0.93	0.93	100.00	1990	0.90	0.90	100.00
Avg.						1	1	95.53	1.25	1.20	97.84		1.16	1.10	97.34		1.20	1.15	97.90

As mentioned in Section II, [13] tested their data augmentation model in the UJIIndoorLoc dataset (UJI 1–2) to improve localisation accuracy. In this case, the authors used the data collected in the first building and second floor. Then, the authors selected 1000 random samples of 1395 from the training dataset, and the remaining samples were added to the testing dataset. As a result, the authors reduced the positioning error by 15.36% using 1000 new positions. Following the same procedure but without reducing the number of APs in the selected data, our approach reduced the positioning error by more than 19% using 8000 new synthetic fingerprints.

2) *Results - M3*: This method provides better performance in terms of floor hit rate than the second method. However, the mean 2D and 3D positioning errors were slightly affected in most of the datasets, for instance, the mean 3D positioning error increased by 6% approximately in DSI 1 dataset (see Table IV).

V. DISCUSSION

In the earlier research, GAN was applied in multiple areas to generate new data that can be pass as real. One of the most representative examples is the generation of artificial faces or pictures which are indistinguishable from real images. Considering this, our proposal is devoted to generating new realistic fingerprints to increase the number of samples in the radio maps and reduce the positioning error.

The new *artificial* fingerprints contain special characteristics that allows to enrich the radio map and reduce the necessity of collecting new fingerprints in the physical location. Additionally, it can be considered as a manner to update the radio map. However, the new fingerprints can be located far from the seed fingerprint or in unreachable indoor areas. We have proposed an algorithm to keep only those new fingerprints that are relevant for indoor positioning (see Algorithm 1).

Additionally, the proposed cGAN architecture allows us to condition the network by adding a label to the data. It diminishes the possibility of having non-relevant fingerprints in the enriched radio map. We found that the set of hyperparameters for the cGAN (e.g., number of epochs or the batch size, among others) is the key to generating unique *artificial* fingerprints.

Given that both the GAN and the positioning model (CNN-LSTM) use convolutional neural networks, it allows extracting important characteristics of the dataset, and even more when the datasets were collected using crowdsourced techniques or when the fingerprints are not collected systematically.

It is also important to select an appropriate distance between the real fingerprints and the *artificial* fingerprints. According to the experiments, if the average distance between points in the dataset is more than 2 m, the distance could be between 1 m to 5 m, but if the average distance is less than 1 m, it is better to choose a smaller distance, for instance, 1 m or less. Due to that cGAN network learns different characteristics of the fingerprints to generate new synthetic data, this may result in a degradation of the accuracy of the classification model. Thus, in multi-floor or multi-building datasets while more *artificial* fingerprints are added to the original dataset less accurate will be the classification model (see Table IV). This issue might be reduced by increasing the complexity of the classification model or using the model trained with the original data.

We can observe the same behaviour in the data augmentation model proposed by [13]; for instance, the performance of the positioning model is better when using 1000 new samples than using 2000. In [13], we can also observe that the authors slightly modify the hyperparameters and/or the number of neurons of the deep neural network implemented to estimate the position. In such a way, the authors achieved similar results with a different number of new synthetic data.

Unlike the previous work [13], we propose a new framework for positioning estimation and data augmentation, which do not require further modification to provide good performance, as shown in Table IV. i.e., our proposed framework requires less fine-tuning effort.

In the same fashion, the accuracy of the proposed framework may vary from one dataset to another (see Table IV). According to the results obtained, we can notice that the accuracy is also related to the number of fingerprints per reference point, the more fingerprints per reference point in the radio map the more the accuracy is improved and vice-versa. i.e., data augmentation works if the radio map in a particular location is rich enough.

VI. CONCLUSIONS

In this research work, we provide a new framework for indoor positioning, which consists of a positioning model CNN-LSTM and a cGAN to generate *artificial* but realistic fingerprints. These two models or architectures are based on deep learning techniques that allow us to extract the most relevant characteristic of the datasets. Thus, the proposed architecture or solution is capable of reducing the positioning error in 70% of the public datasets used in this research work, being the maximum reduction in the positioning error 26% and the minimum 2%.

The solution proposed was tested in seventeen public datasets in order to verify if both models are generalizing well and can be used with multiple Wi-Fi fingerprinting datasets. Furthermore, three methods for training GAN networks have been tested, the first one to train the GAN network by building, being the conditional label the floor, in the second method, the GAN is trained by floor and the last method trains all the dataset without splitting the data. The first and second methods provide better results than the last method. However, all the methods reduced the positioning error with respect to the benchmark.

Future work will analyze new optimization algorithms in order to reduce the instability during the training stage. Also, different variants of GAN and other methods for data augmentation will be tested to provide a complete study of the advantages and disadvantages of each method in relation to fingerprinting indoor positioning.

REFERENCES

- [1] J. Torres-Sospedra, D. Quezada-Gaibor, G. M. Mendoza-Silva, *et al.*, "New cluster selection and fine-grained search for k-means clustering and wi-fi fingerprinting," in *Int. Conf. on Localization and GNSS*, 2020.
- [2] J. Yan, Y. Cao, B. Kang, *et al.*, "An elm-based semi-supervised indoor localization technique with clustering analysis and feature extraction," *IEEE Sensors Journal*, vol. 21, no. 3, pp. 3635–3644, 2021.
- [3] H. lu, G. Xingli, L. shuang, *et al.*, "Indoor positioning technology based on deep neural networks," in *2018 Ubiquitous Positioning, Indoor Navigation and Location-Based Services (UPINLBS)*, 2018.
- [4] D. Asmar, J. Zelek, and S. Abdallah, "Smartslam: Localization and mapping across multi-environments," in *2004 IEEE Int. Conf. on Systems, Man and Cybernetics*, vol. 6, 2004, 5240–5245 vol.6.
- [5] H. Miura, J. Sakamoto, N. Matsuda, *et al.*, "Adequate rssi determination method by making use of svm for indoor localization," in *Knowledge-Based Intelligent Information and Engineering Systems*, vol. 4252, Oct. 2006, pp. 628–636.
- [6] Y. Mao, K. Liu, H. Li, *et al.*, "Scalability of wireless fingerprinting based indoor localization systems," in *2018 15th Annual IEEE International Conference on Sensing, Communication, and Networking*, 2018.
- [7] S. Dai, L. He, and X. Zhang, "Autonomous wifi fingerprinting for indoor localization," in *2020 ACM/IEEE 11th International Conference on Cyber-Physical Systems (ICCPs)*, 2020, pp. 141–150.
- [8] P. Bahl and V. Padmanabhan, "RADAR: an in-building RF-based user location and tracking system," in *Proceedings IEEE INFOCOM 2000*, vol. 2, 2000, 775–784 vol.2.
- [9] X. Song, X. Fan, X. He, *et al.*, "CNNLoc: Deep-Learning Based Indoor Localization with WiFi Fingerprinting," in *2019 IEEE SmartWorld, Ubiquitous Intelligence Computing, Advanced Trusted Computing, Scalable Computing Communications, Cloud Big Data Computing, Internet of People and Smart City Innovation*, 2019, pp. 589–595.
- [10] J. Torres-Sospedra, P. Richter, A. Moreira, *et al.*, "A comprehensive and reproducible comparison of clustering and optimization rules in wi-fi fingerprinting," *IEEE Transactions on Mobile Computing*, 2020.
- [11] S. D. N. Borrás, "Indoor positioning with probabilistic wknn wi-fi fingerprinting," 2016.
- [12] I. H. Alshami, N. A. Ahmad, S. Sahibuddin, *et al.*, "Adaptive indoor positioning model based on wlan-fingerprinting for dynamic and multi-floor environments," *Sensors*, vol. 17, no. 8, 2017.
- [13] W. Njima, M. Chaffi, A. Chorti, *et al.*, "Indoor localization using data augmentation via selective generative adversarial networks," *IEEE Access*, vol. 9, pp. 98 337–98 347, 2021.
- [14] D. Quezada-Gaibor, J. Torres-Sospedra, J. Nurmi, *et al.*, *SURIMI: Supervised Radio Map augmentation with Deep Learning and a Generative Adversarial Network for Fingerprint-based Indoor Positioning – Source code*, <https://github.com/darwinquezada/SURIMI>, 2021.
- [15] M. Radhakrishnan, A. Misra, R. K. Balan, *et al.*, "Smartphones and ble services: Empirical insights," in *2015 IEEE 12th International Conference on Mobile Ad Hoc and Sensor Systems*, 2015, pp. 226–234.
- [16] T.-M. T. Dinh, N.-S. Duong, and K. Sandrasegaran, "Smartphone-based indoor positioning using ble ibeacon and reliable lightweight fingerprint map," *IEEE Sensors Journal*, vol. 20, no. 17, pp. 10 283–10 294, 2020.
- [17] Y. Li, Z. Ghassemlooy, X. Tang, *et al.*, "A vlc smartphone camera based indoor positioning system," *IEEE Photonics Technology Letters*, vol. 30, no. 13, pp. 1171–1174, 2018.
- [18] L. Flueratoru, S. Wehrli, M. Magno, *et al.*, "High-accuracy ranging and localization with ultra-wideband communications for energy-constrained devices," *IEEE Internet of Things Journal*, pp. 1–1, 2021.
- [19] S. Bai, M. Yan, Q. Wan, *et al.*, "DI-rnn: An accurate indoor localization method via double rnns," *IEEE Sensors Journal*, vol. 20, no. 1, pp. 286–295, 2020.
- [20] Z. Chen, M. I. AlHajri, M. Wu, *et al.*, "A novel real-time deep learning approach for indoor localization based on rf environment identification," *IEEE Sensors Letters*, vol. 4, no. 6, pp. 1–4, 2020.
- [21] X. Song, X. Fan, C. Xiang, *et al.*, "A novel convolutional neural network based indoor localization framework with wifi fingerprinting," *IEEE Access*, vol. 7, pp. 110 698–110 709, 2019.
- [22] M. Mirza and S. Osindero, *Conditional generative adversarial nets*, 2014. arXiv: 1411.1784 [cs.LG].
- [23] A. Belmonte-Hernández, G. Hernández-Peñaloza, D. Martín Gutiérrez, *et al.*, "Recurrent model for wireless indoor tracking and positioning recovering using generative networks," *IEEE Sensors Journal*, vol. 20, no. 6, pp. 3356–3365, 2020.
- [24] Q. Li, H. Qu, Z. Liu, *et al.*, "Af-dcgan: Amplitude feature deep convolutional gan for fingerprint construction in indoor localization systems," *IEEE Transactions on Emerging Topics in Computational Intelligence*, vol. 5, no. 3, pp. 468–480, 2021.
- [25] J. Torres-Sospedra, R. Montoliu, S. Trilles Oliver, *et al.*, "Comprehensive analysis of distance and similarity measures for wi-fi fingerprinting indoor positioning systems," *Expert Systems with Applications*, vol. 42, pp. 9263–9278, Dec. 2015.
- [26] B. Xu, N. Wang, T. Chen, *et al.*, *Empirical evaluation of rectified activations in convolutional network*, 2015. arXiv: 1505.00853 [cs.LG].
- [27] G. M. Mendoza-Silva, M. Matey-Sanz, J. Torres-Sospedra, *et al.*, "Ble rssi measurements dataset for research on accurate indoor positioning," *Data*, vol. 4, no. 1, 2019.
- [28] J. Torres-Sospedra, I. Silva, L. Klus, *et al.*, "Towards ubiquitous indoor positioning: Comparing systems across heterogeneous datasets," in *2021 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, 2021, pp. 1–8.