





## Location intelligence system for people estimation in indoor environment during emergency operation

Francesca Fallucchi  [f.fallucchi@unimarconi.it](mailto:f.fallucchi@unimarconi.it)  
 Romeo Giuliano  [r.giuliano@unimarconi.it](mailto:r.giuliano@unimarconi.it)  
 Gianluca Lini  [gianluca.lini@protonmail.com](mailto:gianluca.lini@protonmail.com)  
 Guglielmo Marconi University, Rome, Italy

Alessandro Vizzarri   
 University of Rome Tor Vergata, Rome, Italy  
[alessandro.vizzarri@uniroma2.it](mailto:alessandro.vizzarri@uniroma2.it)

### Abstract

*In the last years, location intelligence systems have been characterized by an increasing interest in several sectors. Among them, those of emergencies are mainly involved in order to enhance the rescue procedures and to reduce the intervention time, especially within indoor environment where GPS does not support the emergency operations. The authors define a low cost location intelligence system based on Channel State Information (CSI) of Wi-Fi and low-energy ESP32 SoC platform to analyze CSI data of Wi-Fi Signals. The technical solution utilizes wavelet filter to remove background noise in the CSI data, Principal component analysis (PCA) to reduce the dimensionality of the CSI data and get the most valuable data that are used as feature for the defined DNN model. The experimental results show the best performance of this model compared to the other machine learning (ML) algorithms analysed.*

### 1. Introduction

Location intelligence is an ever-growing topic and in the last years has gained the interest. According to the “location Intelligence - Global Market Trajectory & Analytics” report, although the COVID-19 crisis, the global market of location Intelligence is estimated at \$11.9 Billion in 2020 and is forecast to grow with a CAGR of 14% over the 2020-2027 period, reaching \$29.8 Billion by 2027[1]. Also from a scientific point of view, location intelligence has obtained a growing interest from the scientific community due to the various applications all focused on the importance of data and information related to location. In particular, for emergency situations in indoor environments it becomes of crucial importance, given that location systems based on positioning satellites are not available. The possibility of having information on the position of people to be rescued in indoor environments is essential for rescuers, especially for the first responders, who are the first rescuers to intervene in the emergency

context. If this information arrives in a very short time, it allows to increase the intervention times and to save human lives, minimizing the negative consequences of the emergency event. Actually, many location systems need a huge amount of preliminary activities to perform in order to collect data, to train them and then to proceed with their processing. Moreover, dedicated equipment to be deployed are also a crucial issue both from an operative, maintenance and cost point of view.

In this paper, a simple low-cost location system for emergency in indoor environment is proposed, where the first responders may need to find persons within rooms inside a large building. Usually, it is not possible to check in brief time the presence of people in each room of the building and for each floor. To solve this problem, a location system of persons in an indoor emergency situation is defined. It is not necessary that persons are equipped with any personal device. The proposed system is based on the measures collected by the local Wi-Fi networks already deployed within building, offices, hotels, and houses. During the emergency phase, a local station (STA) can start to send specific packet to the Access Point (AP). If a person is present in the room, he interacts with the electromagnetic field, thus modifying the response received by the AP. The AP collects the packets and then sends them to the server equipped by a neural network, which can predict if a person is present in the room or not. This can help the first responder to speed up the person detection procedure in the building and then to reduce the time necessary to rescue him.

The main paper’s contribution consists of three aspects: (i) development of a neural network model for human detection where the performances obtained are compared with other machine learning algorithms known in the literature; (ii) network performance analysis in terms of transmission parameter settings (regarding the number of packets analyzed), the location and number of human detection devices (Active AP and Active STA) for data collection and extension of the neural network model to a second room; (iii) and an

efficient integration of the local communication network with the rescuer's proprietary communication network is proposed by the authors. In order to reduce the end-to-end (e2e) delay, different levels of integration are proposed between the two networks considered (the local network for human detection and the proprietary rescuer network), including the use of Edge Cloud Computing (ECC) platforms. The performances obtained in the different integration strategies are then analyzed and compared. The research presents several strengths. The experimental tests have been carried out using two couples of sensors in a separated way in order to extend the neural network input. Moreover, the analysis of WiFi CSI data constitutes a known approach according to the scientific literature. The identification of a low-cost network to be used by the firefighters and to integrate it with the emergency private network (as PMR) is very important in order to offer other connections to the server.

The weaknesses of the research mainly regard that the experimental tests have been only carried out in a single apartment. A final data validation should be also performed in other indoor environments. The feature extraction was performed before the DNN model to better validate the signal quality offered by low cost devices. Finally, experimental tests for different server deployment strategies should be useful for a performance comparison of the different scenarios.

The paper is organized as follows: Section 2 presents the main related work in the field of people estimation using radio signals. Section 3 describes the overall system and the main functionalities. Section 4 describe the workflow defined for data processing and preparation. It also define the Machine Learning model used for prediction. Section 5 illustrates the carried out experimentation describing the overall set up and equipment deployment. Section 6 summarizes the main obtained results. Finally, the Section 7 draws the main conclusions and the future works.

## 2. Related works

The signal strength refers to the transmitter output power as received by a reference antenna at a distance from the transmitting antenna. The Received Signal Strength Indicator (RSSI) is a measurement of the power carried out by a received radio signal. In this way using one or more TX station and one or more RX station, the research can try to identify an appropriate model for people estimation analyzing the RSSI value received at RX related to the number of people. Researchers have studied the property of radio signal strength changing in the people areas.

By using two different statistical methods, Fadhlullah and Ismail [2] investigated the differences in signal attenuation between dynamic and static people and they also searched for significant people properties that affect wireless signal propagation. Their proposed system used 2.4 GHz ZigBee node and it was able to estimate with 75.00% and 70.83% accuracy the low and medium human people densities, respectively. Ding et al.[3] estimated the people density using passive RFID tag. Their approach is based on the variance of RSSI values of the receiving signal. Accordingly, they deployed around 20 RFID passive tags in the target area and utilised the RFID reader to estimate the people density based on analysing the backscattered signal from the passive tags. Yuan et al. [4] proposed an RSSI device-free people counting approach, which exploits the space-time relativity of people distribution in order to minimise the estimation errors. In a recent work of Taha et al. [5] people is estimated for a given measured RSSI based on the trained kNN classifier. They use one transmitting and two receiving wireless nodes for training the classifier. Radio Frequency data is collected to build up a fingerprint database consisting of features. Regarding the accurate feature extraction and optimization of the time required for training Bootstrap re-sampling technique has been used. They stated that presented kNN classifier has shown 94% accuracy using n-fold cross validation for up to 14 people people density. Due to its diffusion and its easiness in managing control signals, Wi-Fi Channel State Information (CSI) signals are commonly used in researches on current human sensing. Halperin [6] in 2011 was one of the first to release a tool for gathering CSI from 802.11n signals using common Intel wireless card using Linux SO. Project's website [7] is up to date with a growing list of publication that cite this work and use the tool for research purpose. Xie [8] realized another tool based on Atheros chipset, also this one has a up to date site[9]. There are many interesting works on Wi-Fi CSI data. For example Di Domenico et al. [10] performs Doppler Spectrum transformation to a CSI stream. Groups of people with 0 to 8 participants were measured in 3 different locations. The authors selected Spectral Kurtosis as the unique descriptor for their model and for the learning stage. In this case, the authors used a Naive Bayes classifier, with this setup, they achieve about 80% of accuracy for people counting estimation. Liu et al. proposed two different methods based on deep learning. WiCount [11] use deep learning approaches to construct correlation between the number of people and CSI values. The prototype of WiCount is implemented and evaluated on the commercial Wi-Fi device. Proposed solution is able to estimate the number

of people up to 5 with the accuracy of 82.3%. With DeepCount [12] they proposed a method that use deep learning techniques. In particular, the Convolutional Neural Network (CNN) to extract channel and number of people relationship feature and Long Short-Term Memory (LSTM) to resolve the dependencies of number of people and CSI. According to the authors with massive training samples they achieve an accuracy of 86.4% in an environment of up to 5 people. Xi[13] present a device-Free people Counting approach based on CSI. He introduces a feature called Percentage of non-zero elements (PEM), which is a measurement technique based on the non-zero counting of the CSI dilated matrix. Using Grey Verhulst Model he was able to express a relationship between CSI data and people without a labor-intensive site survey. Performance of their method was evaluated utilizing the probability that an error equal or less than a defined threshold occurs for a particular counting estimation. This indicator was reported to be 98% for an error lower than 3 persons and about 80% for an error of 1 person. A recent survey on Wi-Fi sensing using CSI feature could be found in Ma's work [14].

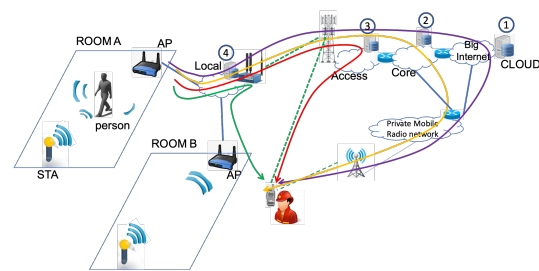
### 3. Overall system and its working

The collection and elaboration station could be a small pc that acts as a data gateway and forward the real-time collection or a could be an edge computing solution that record and elaborate the data in place. The proposed system is applied to the following emergency use case in order to reduce the intervention times by rescuers. During an emergency within a large building, the firefighter has to find out if someone is still inside the building (e.g., on fire) in the most reduced time. He exploits the proposed location intelligence system by connecting to its server. The overall system starts to collect measures for the prediction of possible people within rooms of the building. Measures are sent to the server where the neural network is running. Based on the given elaboration of the neural network, the server provides the first responder with the information of rooms where people can still stay. Depending on where the server is located within the network, the firefighter can use his private mobile radio network or Private Mobile Radio (PRM), which is dedicated to civil protection and other public national forces such as police, firefighters or soldiers, the public mobile cellular network or the local Wi-Fi. The WiFi AP cannot be the same operating in the rooms in order to avoid network connection issues due to walls or obstacles. We analyzed four cases:

1. The server is located in the Internet cloud.

2. The server is located inside the Telecom Provider network.
3. The server is located in the Multi-access Edge Computing (MEC) within the Telecom Provider network.
4. The server is locally deployed and the access occurs through a local wireless connection such as Wi-Fi.

In Figure 1 the position of the server within the network is depicted. The envisaged links between the considered data path for the collected measures is evidenced, together with the position of the server where the neural network runs and the subsequent information sent to the first responder on the possible room where a person can still be located.

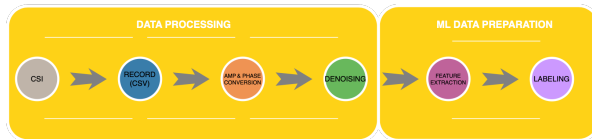


**Figure 1. Envisaged strategies for deploying the server within a network.**

In the monitored environment (such as hotel, house or office room), a small Local Collecting Network (LCN) is used to collect data. The main component used to build the LCN are the following:

1. Active Station (namely Active STA): used for active CSI exchange. It connects to some Access Point (AP), in the considered case another ESP32, and sends packet requests that contain CSI information.
2. Active AP: Active CSI collection AP. It's another ESP32 acting as AP to be connected to by STA. It collects STA CSI data and respond to packet request with reply that also contains CSI data.

The Active Station sends packets to the Active AP collecting CSI data within the room in order to characterize the indoor environment. If a person is present or not in the room, the CSI information collected by ESP32 devices (Active AP) can be different due to the modification of electromagnetic field. In this way, the response received by the AP is quite different in case of the presence or absence of a person in the considered indoor environment. Since the devices are connected



**Figure 2. Data Processing and Data Preparation for ML algorithm.**

through a USB-UART adapter, when the Active AP receives the WiFi CSI information from the Active Station, it forwards data to the collection and elaboration station, where a pre-processing is done to prepare data for the Machine Learning model.

#### 4. ML model with LCN data

The research is based on Wi-Fi signals for CSI data collection. Regarding the LCN structure The ESP32 platform is identified, specifically ESP32-D0WD-based modules with integrated flash. These modules are equipped with Wi-Fi and Bluetooth/Bluetooth LE-based connectivity and a dual-core CPU. Since the obtained platform is cheap and portable with a native support for CSI to be extended to other Wi-Fi sensing problem, and also it could allow some hardware hacks, it can be used for future research topics. It also enables the usage of more than a single couple of transmitter/receiver to build different dataset. Two recent works have exploited ESP32 Wi-Fi CSI capabilities, Atif et al [15] proposed Wi-ESP as a tools for CSI measurement and Device-Free Wi-Fi Sensing (DFWS). The present paper evaluates the performance of ESP32 from a signal perspective and it represents a starting point useful for next research topics, although more information need to be clarified yet, such as the statement about the non-randomness of phase value collected by Wi-ESP. Hernandez et al. [16] proposed ESP32 CSI Toolkit<sup>1</sup> as a way to give to researchers the access to CSI directly from the ESP32 microcontroller. The present work started with a customization on ESP32 CSI toolkit software code to allow the compiling with Espressif IoT Development Framework (ESP-IDF) v4.x instead of the original requirement of ESP-IDF's version v3.3.1<sup>2</sup>. The adopted workflow can be summarized in the following steps: (i) collection of csi data and recording on csv, (ii) transformation of raw values in amplitude and phase, (iii) filtering, (iv) dimensional reduction and labeling. The whole processing is illustrated in Figure 2.

<sup>1</sup><https://stevenmhernandez.github.io/ESP32-CSI-Tool/>

<sup>2</sup><https://docs.espressif.com/projects/esp-idf/en/latest/esp32/api-guides/wifi.html#wi-fi-channel-state-information>

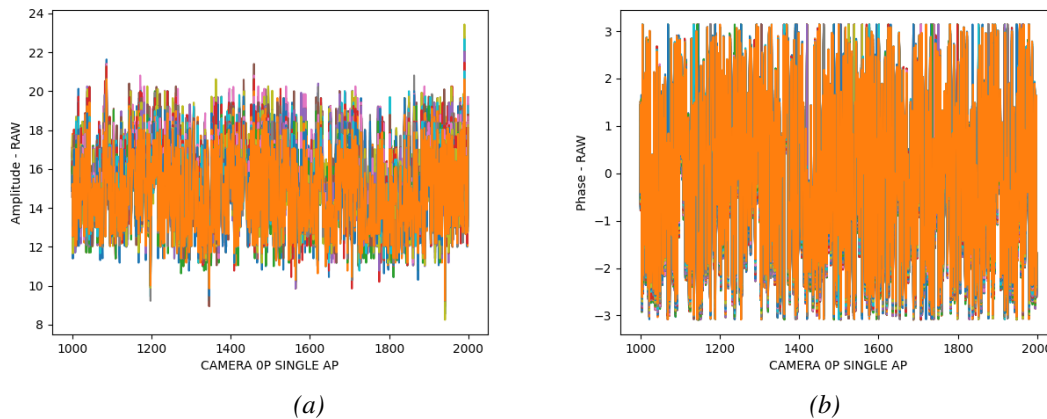
#### 4.1. Data Processing

A file in csv format resume the raw data collected from the ESP32 and can be used in the CSI conversion and filtering phases.

The file contains many fields like the RSSI of packet, Signal Mode for determine packet type (non HT(11bg)/HT(11n)/VHT(11ac)), Channel Bandwidth (20MHz/40MHz) and a vector of RAW CSI data collected during the experimental phase. In this phase (data exploration) all data are collected. Default mode for ESP32 is HT40. However, the actual bandwidth is firstly negotiated during the Wi-Fi connection. It is HT40 only if both the station and the connected AP support HT40, otherwise it is HT20. If the bandwidth of connected AP changes, the actual bandwidth is negotiated again without Wi-Fi disconnecting. In the carried out experiment a small variation on channel bandwidth with few HT20 packets was observed. It means a low rate of renegotiation was observed in the test environment. Recent Wi-Fi standard uses OFDM at the physical layer and follow the *IEEE802.11n/ac* standard. It allows multiple TX and RX antennas for MIMO communication. The amplitude and phase of CSI are affected when there are physical changes in the surrounding environment[17]. To calculate CSI, the Wi-Fi TX assigns long training symbols (LTFs), which contains in the packet preamble the pre-defined symbols. When the LTFs are obtained, the Wi-Fi RX calculates the CSI matrix using the received signals and the original LTFs. For each subcarrier, the Wi-Fi

channel is designed by equation:  $\bar{y} = \bar{H} \cdot \bar{x} + \bar{n}$  where y is the signal received by RX, x is the signal transmitted from TX, H is the CSI matrix, and n is the noise vector. After receiving y, the RX calculates the CSI matrix H using x and the received signal y after eliminating the cyclic prefix, remapping, and OFDM demodulation. The final estimated CSI is a 3D matrix of complex values, which evaluates the channel quality using each entry of the matrix. Final CSI values[14] could be represented with the following

equation:  $H_i = |H_i|e^{j\angle H_i}$  where  $H_i$  and  $\angle H_i$  are the amplitude and phase response of the i-th subcarrier, respectively. This fine-grained information was then converted to amplitude and phase mapping the value of the received raw CSI value array accordingly. More info on the OFDM modulation and the ESP32 could be found in Atif work [15]. In this phase of data processing, Python code for process CSV data with Pandas library for data analysis was used. A preliminary analysis of the collected data from the ESP32 is performed to identify the best strategy for post signal processing. After a first



**Figure 3. (a) Raw amplitude data graph over a 1000 CSI packet samples collected in a room with no people, (b) Raw phase data over a 1000 CSI packet samples collected in a room with no people.**

data visualization (see Figure 3) the signal processing procedure became necessary especially from a raw signal denoising point of view. Initially, the usage of a Butterworth filter is analysed. It is a signal processing filter designed to have a frequency response as flat as possible in the passband. It was first described in 1930 by the British engineer and physicist Butterworth [18] and it's widely used in digital signal processing [19, 20]. The Gaussian filter is also considered to denoise the signal [21]. The better result was obtained adopting the wavelets filtering methods[22, 23], in particular the Daubechies wavelets. [24] of 4th order or db4 with PyWavelet[25] library. Daubechies 4th order filter was used and the resulting low pass filter matrix value was considered as CSI filtered source value to denoise the signal. After the filtering, the signal is slightly smoothed and with less spike. This data could be used as input of a ML/DL algorithm but in this paper we decide to do some data preparation to reduce the dimensionality.

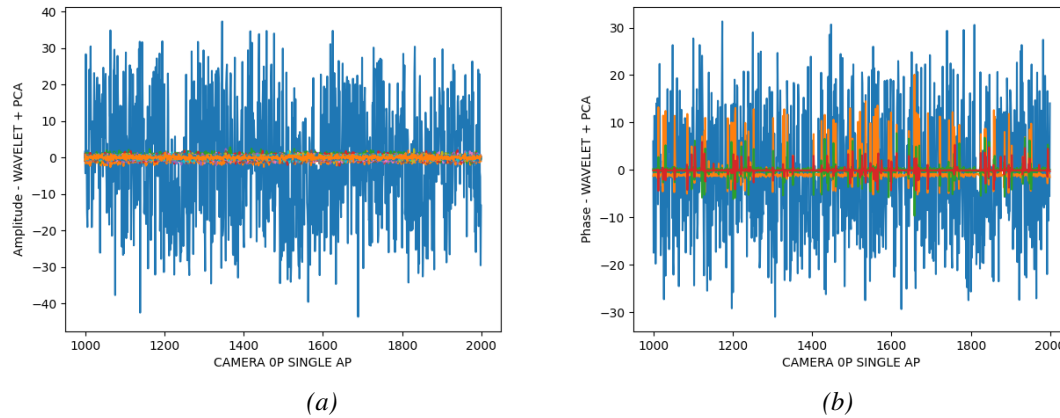
## 4.2. ML data preparation

Some dimensionality reduction functions should reduce the complexity of a DNN exploiting built-in capability of feature extraction in this first scenario. Reducing the number of variables allows a reduction of input value in the DNN from 104 to less (16/32 in the considered case). PCA is the best scientific methodology to optimal number of variables that describe the collected data without loose data meaning. PCA is the process of computing the principal components and using them to perform a change of basis on the data, sometimes using only the first few principal components and ignoring the rest. The principal components are eigenvectors of the data's

covariance matrix. Thus, the principal components are often computed by eigen-decomposition of the data covariance matrix or singular value decomposition (SVD) of the data matrix. After fixing a threshold of the variance, the number of principal components that represent well the dataset is identified. It allows to decompose the data and obtain a small number of feature input in the considered dataset. The experiment shows that with a variance between 0.96-0.98 good numbers are 12 for data reduction of the amplitude matrix and 4 for data reduction of the phase matrix. Using 12 and 4 as output of PCA grant us some linearity in the feature expansion without growing too much the ANN. The final result after the PCA can be seen in Figure 4.

After the PCA the graph appears different from the previous one, in particular the phase graph seems usable as a feature input for some machine learning (ML) algorithm. The last step is data labeling, it is basically adding one (or more) informative labels to the collected data in order to provide context so that a ML model can learn from it [26, 27, 28, 29]. In supervised learning, a labeled set of data is needed, so that the model can learn from to make correct decisions. The tagging in the considered case is a people class tag that the ML model use to learn the underlying patterns in a process called "model training". The result is a trained model that can be used to make predictions on test data. A properly labeled dataset provides a ground truth that the ML model uses to check its predictions for ac-curacy and to continue refining its algorithm. The accuracy of trained model will depend on the accuracy of the ground truth, so spending the time and resource to create a good dataset is essential. The considered case could be treated as a classification problem. In this case, the labels  $y_i$  are taken from a set of possible classes. Given training





**Figure 4. (a) 12 PCA Amplitude data graph over a 1000 CSI packet samples collected in a room with no people. (b) 4 PCA phase data over a 1000 CSI packet samples collected in a room with no people.**

data consisting of a set of pairs  $(x_i, y_i)$  of samples  $x_i$  and their labels  $y_i$ . A model is learned that can predict the label  $y_j$  for a previously unseen sample  $x_j$ . In the people estimation from CSI value problem, these classes would be  $Y = 0P, 1P, 2P, 3P, 4P, \dots, nP$ . The model sought of a machine learning system in this case can be equated again to a function. The function value  $f(x_j)$  is the predicted class of  $x_j$ . There are many algorithms that can be used as supervised learning, the most common are Naïve Bayes, Support Vector Machine (SVM), K-nearest neighbor algorithm, Neural Networks (NN) also called Artificial Neural Networks (ANNs). Deep Learning (DL) is part of a broader family of machine learning methods based on ANNs, the deep word in DL refers to the use of multiple layers in the network. Algorithms known in literature are used to compare the result of defined model based on DL. All the models are built in Python with the Keras framework. Keras is a deep learning API, it is a powerful and easy-to-use free open-source Python library for developing and evaluating deep learning models. It wraps the efficient numerical computation libraries TensorFlow and allows to define and train neural network models. The goal here is to find if with the proposed approach can solve the problem of correctly predict the people in a room – and maybe in other room without retrain the model – without constrain on performance or code.

### 4.3. Prediction

Data labeling is the last step involved in data preparation for training and testing of the defined ML model. To correctly predict the people in a room, different models were analyzed. Finally, the modelled reference model for the evaluation of CSI data is a

deep neural network with a variable input layer, varying from 16 to 32 in the adopted solution but more info on that later, 10 fully connected hidden layer with  $4 \times 100$ ,  $4 \times 50$  and  $2 \times 25$  nodes, an output layer with 3 nodes for match the class (see Figure 5). With 32 input 48303 trainable parameters are obtained for fine model tuning. For the inner (hidden) layer, the ReLU (Rectified Linear Unit) is used as the activation function. It is a non-linear function that gives the same output as input if the input is above 0, otherwise the output will be 0. The ReLU function also helps with the problem of vanishing gradients in deep networks by not squashing in both ends. The output layer uses a softmax function for deal with the multi-classification problem. It highlights the largest values and suppresses values which are significantly below the maximum value, though this is not true for small values. Out is normalized so that they sum to 1 and they can be directly treated as probabilities over the output. In the considered classification problem, the predicted probability  $P(x)$  is obtained for every class  $x$  that is used in testing phase for model evaluation. In the defined DNN model, the categorical\_crossentropy was used as loss function and Adam [30, 31] as the optimizer for ANNs training. Categorical\_crossentropy is a loss function that is used in multi-class classification tasks. These are tasks where an example can only belong to one out of many possible categories, and the model must decide which one. Formally, it is designed to quantify the difference between two probability distributions. Adam [30, 31] is the go-to-optimizer. It efficiently computes according to stochastic gradient descent-methods. Adam can be viewed as a combination of RMSprop and momentum. Momentum can be seen as a ball rolling down a slope, Adam behaves like a heavy ball with friction, which thus

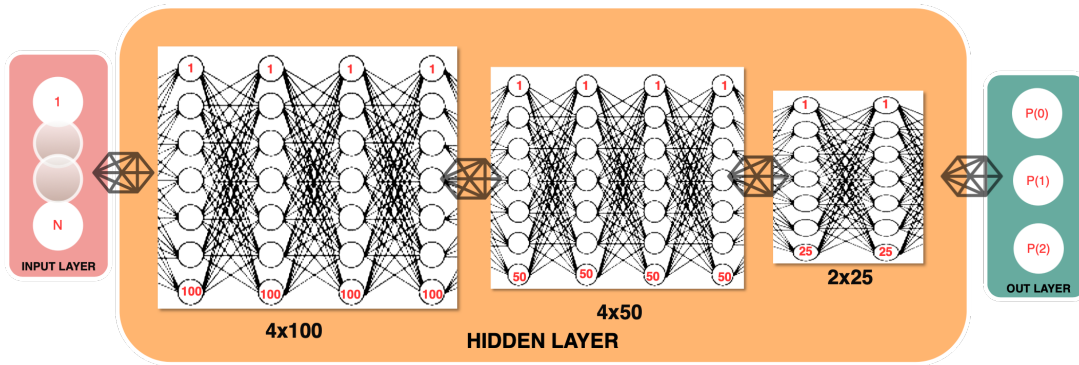


Figure 5. The proposed DNN with N input layer (where N is 16 or 32 nodes according to Table 1), 10 hidden layer with 4x100, 4x50 and 2x25 nodes and 3 nodes for the output layer.

prefers flat minima in the loss function landscape. The authors of the original paper empirically showed that Adam works well in practice and compares favorably to other adaptive learning-method algorithms. RMSprop contributes the exponentially decaying average of past squared gradients, while momentum accounts for the exponentially decaying average of past gradients.

## 5. Experimentation

After the signal analysis phase and the creation of the data workflow (Figure 2) to build the dataset, the test scenarios was defined. Two rooms were identified, where the CSI data were collected. The rooms are those of an inhabited house - so a real scenario with furniture in the rooms and neighbors that might have Wi-Fi devices - and we took the CSI data in three scenarios: 0 people (empty room), 1 person, 2 people. This led to identify three classes of membership for labeling the data accordingly, so the class data Y will be  $Y = 0P, 1P, 2P$ . Three different experiments (D1, D2 and D3) were carried out using several Active STAs / Active APs. The following Table 1 summarizes the main characteristics of them:

Table 1. Summary of experimental setup

Scenario	Sensor Couple	Position	Feature (Input)
D1	1	parallel(=)	16
D2	2	cross(+)	32
D3	2	parallel(=)	32

The first experiment (D1) involved the use of a pair of ESP32 acting as an STA and Active AP couple placed in parallel in the room to create the measurement signal and evaluate its change as a function of the number of people present in the room. The positioning was done on the room diagonal, specifically from the bottom

left vertex to the top right vertex. ESP32 sensors are positioned at a height of about 1m from the ground. In the ROOM A we have carried out two independent measurements, the first one was collected to create the training dataset and the second one in order to create the test dataset. In ROOM B we have carried out a single measurement that has been used in order to verify if the model could be exportable also in other environments without carrying out a retraining of the net. We wanted to verify the effectiveness of the model in similar environments - house - to see if the research idea could have some real-world scenario use. CSI data collected from ESP32 generate 104 value for packet. Those value are equally split between Amplitude and Phase. Signal analysis made us understand the importance of the reduction of dimension (PCA). This led us to define the input layer of defined DNN for this scenario as consisting of 16 inputs, 12 are the amplitude values and the other 4 are the phase values, both characterized well the collect data after the PCA reduction.

The second round of test is two couple of ESP32 and came after some reason on the result obtained from the first scenario. The idea here is to evaluate if increasing the number of sensors could lead to an increase in prediction. In this second scenario (D2) the idea was to simulate a MIMO system using two pairs of ESP32. Each pair of ESP32 active\_sta/active\_ap had its own independent Wi-Fi signal so that the active\_sta had its active\_ap of reference and therefore two different wireless networks are available on the room. The positioning of the sensors in the room was in the shape of a cross, sensors are placed in the middle of each wall at a height of about one meter from the ground with the pairs of active\_apactive\_sta are perpendicular to each other. The first idea was to create a larger dataset in terms of size in order to have more training data. It could be done by temporally aligning the data collected independently from the two sensors and going

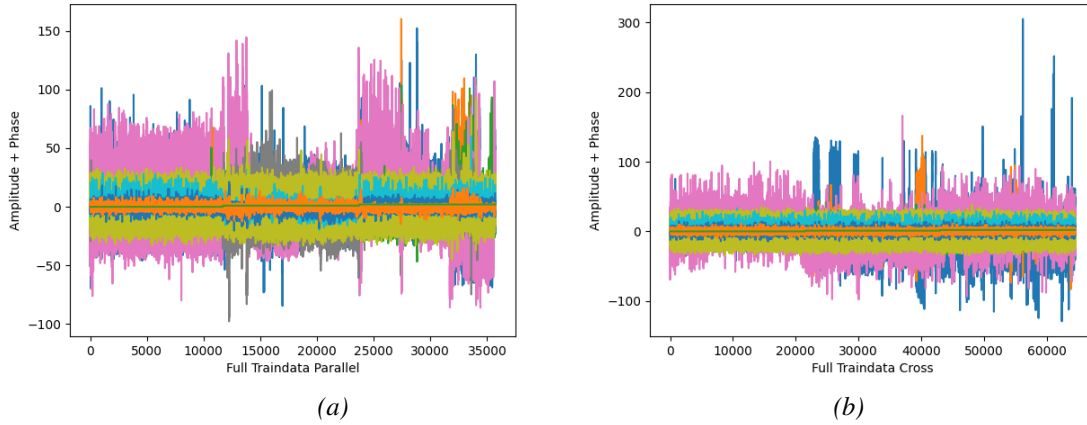


Figure 6. Graph on full training data (a) with sensor in parallel and (b) with cross positioning

to increase the database. Instead, we have assumed a feature expansion for the network input. Also, in this case two independent measurements were collected for training and testing in ROOM A and one for testing in ROOM B.

In the third scenario (D3) the basic idea was always to simulate a MIMO with two pairs of ESP32as the previous one but this time the positioning was in parallel, with both sensor on the diagonal of the room as in the first scenario. We test the model using DNN input with feature expansion. Also, in this case two independent measurements were collected for training and testing in ROOM A and one for testing in ROOM B.

### 5.1. Feature expansion and data measurement elaboration

Each pair of ESP32 STA/Active\_AP is treated in an as independent. A time alignment is performed between the two recorded tracks, and then we apply the basic workflow for signal processing. In the considered case, the amplitude and phase values are 12 and 4 respectively for each ESP32 pair. The result of each processing is then expanded bringing the neural network inputs to 32. The expansion is not vertical (it means the number of records in the dataset did not increased), but it is horizontal. The input features of considered network increased so the input data is represented by matrix of n row with n as number of training or test sample and 33 columns, 32 are the input of the DNN and one is the label class. In Figure 6 there are some graphs of the whole training dataset in both of the above mentioned experiment.

### 5.2. Test data compression and collected data

During the model evaluation phase, a real use scenario of the system have been identified. Since the people estimation in a room with a resolution of less than a second is quite difficult to implement, the adopted solution can treat the collected data in a weighted manner. In the data compression, the median of CSI value for subcarrier within a variable is calculated. It is a configurable number of packets that constitute the testing dataset. Tests were performed with 1 or no filtering, 5, 10, 50 packets to find that good values are 5 or 10. During the measurement people were free to move in the room when the signal was recorded, so there is no static position enforced for the user. Each test had a duration from two to three minutes. The data collection rate was about 200 packet/s. This led to a single csv data file of around 20MB/measure for every CSI class. In Table 2 the training data-set are reported and in Table 3 the testing data-set used for the model evaluation is summarized, together with the conclusion. The training dataset was constructed from equally distributed packet sample between labelling classes 0P, 1P, 2P.

Table 2. Summary of Training Dataset

Scenario	Sensor Couple	Position	Train [pkt]
D1	1	parallel(=)	38890
D2	2	cross(+)	64509
D3	2	parallel(=)	35726

## 6. Results

In this section, some results are reported together with a comparison about the considered model and some



**Table 3. Summary of Testing Dataset**

Scenario	Sensor Couple	Position	Class	Test[pkt]
D1	1	parallel(=)	0P, 1P, 2P	7917
D2	2	cross(+)	0P	14053
D2	2	cross(+)	1P	16420
D2	2	cross(+)	2P	15753
D3	2	parallel(=)	0P	11700
D3	2	parallel(=)	1P, 2P	12013

well-known ML algorithm (see Table 4). The mentioned results are only referred to a subset of carried out tests, but they are useful for the final conclusions. Table 4 reports the prediction of the considered DNN system in the different scenario listed in Section 5 for the 0p class.

**Table 4. Comparison about the defined model and some well-knowman ML algorithm on 0P class.**

Scenarios	Model	Test Case	Ex. Pred.
D1	DNN	ROOM A – 0P	87,16
		ROOM B – 0P	71,86
	SVM	ROOM A – 0P	96,88
		ROOM B – 0P	78,86
D2	DNN	ROOM A – 0P	78,93
		ROOM B – 0P	37,96
	KNN-5	ROOM A – 0P	95,27
		ROOM B – 0P	67,88
	N-BAYES	ROOM A – 0P	98,13
		ROOM B – 0P	79,62
	SVM	ROOM A – 0P	95,77
		ROOM B – 0P	68,15
D3	DNN	ROOM A – 0P	99,35
		ROOM B – 0P	78,63
	KNN-5	ROOM A – 0P	97,18
		ROOM B – 0P	78,78
	N-BAYES	ROOM A – 0P	95,43
		ROOM B – 0P	91,59
	SVM	ROOM A – 0P	97,78
		ROOM B – 0P	85,44

Figure 7 (a), shows a comparison with other model on ROOM A for class 1p and 2p (presence of people in the room). The D3 scenario was the one capable of better results also with other ML algorithms.

The defined model was also used to perform other tests in a second room (Room B) and to collect other data, different from those collected in the first room (Room A). In Figure 7 (b), a comparison between the defined model and other ML algorithms is shown. In the graph the same pattern observed in ROOM A is

obtained. D3 scenario is the one that perform better and the adopted DNN model shows good results.

## 7. Conclusion and Future Work

The DNN model defined by the authors offers results very important for a rescue point of view in an indoor emergency situation. Although some classical ML models have good results, the defined neural network in many cases performs better, the model behaves quite well even with signals collected in environments different from the training one. However more tests in different rooms, maybe smaller than the one used or even larger, would be necessary for further validation of the model. Gathering a high-quality dataset is the most important part of the entire machine learning process, especially in supervised learning. However, preparing a complex dataset is a lot of intense work. The necessity to have an efficient signal processing strategy is to be underlined, since it allow to reduce the complexity of the ANN model.

Regarding the future works, the authors identify the following possible actions. A real-time workflow should be defined and implemented in order to test the model. It means the CSI gathered data will be processed and evaluated on the fly by the model. To this purpose, EspressoIF recently proposed for the Chinese market a solution called ESP Lamp that does something similar [32]. The support of multiple antennas should be also evaluated, since ESP32 device supports up to sixteen antennas through external antenna switch. The feature extraction from CSI data can be performed by a neural network (Encoder/CNN/LSTM) instead of using PCA for dimensional reduction. Finally, a comparison analysis of the defined DNN model can be carried out considering other hardware platforms, like some cots Wi-Fi AP with built-in Atheros or Intel chipset that support CSI extraction.

## References

- [1] "Location intelligence - global market trajectory analytics, id: 5302503 report april 2021 region: Global 302 pages global industry analysts, inc." <https://www.researchandmarkets.com/r/9xly8a>, accessed Apr. 24, 2021. Available on line.
- [2] S. Fadhlullah and W. Ismail, "A statistical approach in designing an rf-based human crowd density estimation system," *International Journal of Distributed Sensor Networks*, vol. 2016, pp. 1–9, 02 2016.
- [3] H. Ding, J. Han, A. X. Liu, J. Zhao, P. Yang, W. Xi, and Z. Jiang, "Human object density estimation via backscattered radio frequency signal," in *2015 IEEE Conference on Computer Communications (INFOCOM)*, pp. 1652–1660, 2015.
- [4] Y. Yuan, J. Zhao, C. Qiu, and W. Xi, "Estimating crowd

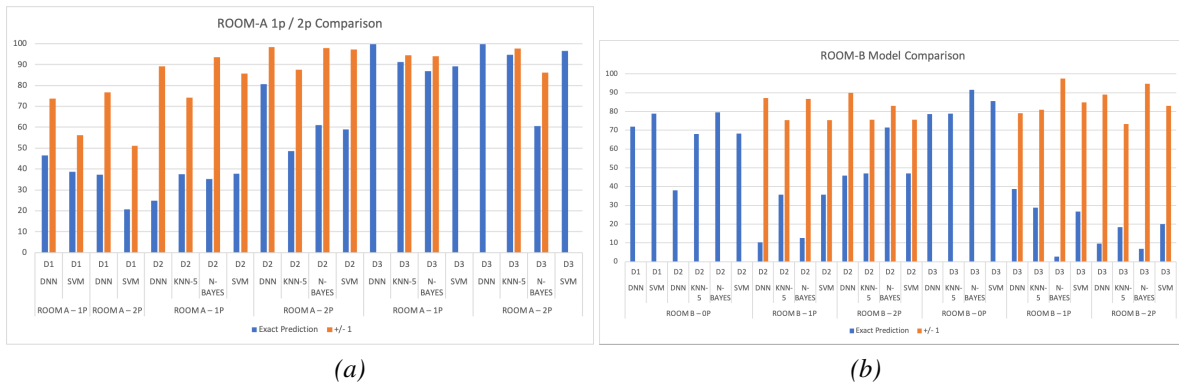


Figure 7. (a) ROOM A Model Comparison (b) ROOM B Model Comparison

- density in an rf-based dynamic environment,” *IEEE Sensors Journal*, vol. 13, no. 10, pp. 3837–3845, 2013.
- [5] M. Taha and et.al, “Crowd estimation based on rssi measurements using knn classification,” in *3rd Int. Conf. Int. Aut. Sys. (ICoIAS)*, pp. 67–70, 2020.
- [6] D. Halperin, W. Hu, A. Sheth, and D. Wetherall, “Tool release: Gathering 802.11n traces with channel state information,” *Computer Communication Review*, vol. 41, p. 53, 01 2011.
- [7] “Linux 80211n csitool.” <https://dhalperi.github.io/linux-80211n-csitool/>, accessed Apr. 24, 2021. Available on line.
- [8] Y. Xie, Z. Li, and M. Li, “Precise power delay profiling with commodity wi-fi,” *IEEE Transactions on Mobile Computing*, vol. 18, no. 6, pp. 1342–1355, 2019.
- [9] “Atheros csi tools.” <https://wands.sg/research/wifi/AtherosCSI/>, accessed Apr. 24, 2021. Available on line.
- [10] S. Di Domenico, G. Pecoraro, E. Cianca, and M. De Sanctis, “Trained-once device-free crowd counting and occupancy estimation using wifi: A doppler spectrum based approach,” in *IEEE 12th Int. Conf. Wirel. Mob. Comp., Netw. and Comm. (WiMob)*, pp. 1–8, 2016.
- [11] S. Liu, Y. Zhao, and B. Chen, “Wicount: A deep learning approach for crowd counting using wifi signals,” in *IEEE Int. Sym. Parallel and Distributed Processing with App. and IEEE Int. Conf. on Ubiquitous Comp. and Comm. (ISPA/IUCC)*, pp. 967–974, 2017.
- [12] S. Liu, Y. Zhao, F. Xue, B. Chen, and X. Chen, “Deepcount: Crowd counting with wifi via deep learning,” 2019.
- [13] W. Xi, J. Zhao, X.-Y. Li, K. Zhao, S. Tang, X. Liu, and Z. Jiang, “Electronic frog eye: Counting crowd using wifi,” in *IEEE INFOCOM 2014 - IEEE Conference on Computer Communications*, pp. 361–369, 2014.
- [14] Y. Ma, G. Zhou, and S. Wang, “Wifi sensing with channel state information: A survey,” vol. 52, no. 3, 2019.
- [15] M. Atif, S. Muralidharan, H. Ko, and B. Yoo, “Wi-ESP—A tool for CSI-based Device-Free Wi-Fi Sensing (DFWS),” *Journal of Computational Design and Engineering*, vol. 7, pp. 644–656, 05 2020.
- [16] S. M. Hernandez and E. Bulut, “Lightweight and standalone iot based wifi sensing for active repositioning and mobility,” in *2020 IEEE 21st International Symposium on “A World of Wireless, Mobile and Multimedia Networks” (WoWMoM)*, pp. 277–286, 2020.
- [17] D. Zhang, H. Wang, and D. Wu, “Toward centimeter-scale human activity sensing with wi-fi signals,” *Computer*, vol. 50, no. 1, pp. 48–57, 2017.
- [18] S. Butterworth, “On the theory of filter amplifiers,” *Experimental Wir. and the Wir. Engineer*, vol. 7, 1930.
- [19] I. Selesnick and C. Burrus, “Generalized digital butterworth filter design,” *IEEE Transactions on Signal Processing*, vol. 46, no. 6, pp. 1688–1694, 1998.
- [20] L. Zhongshen, “Design and analysis of improved butterworth low pass filter,” in *2007 8th International Conference on Electronic Measurement and Instruments*, pp. 1–729–1–732, 2007.
- [21] S. Karmakar, K. Ghosh, R. Saha, S. Sarkar, and S. Sen, “A new design of low pass filter by gaussian derivative family,” in *2005 3rd International Conference on Intelligent Sensing and Information Processing*, pp. 177–182, 2005.
- [22] G. Luo and D. Zhang, *Wavelet Denoising*. 04 2012.
- [23] P. Dautov and M. S. Özerdem, “Wavelet transform and signal denoising using wavelet method,” in *2018 26th Signal Processing and Communications Applications Conference (SIU)*, pp. 1–4, 2018.
- [24] *DAUBECHIES WAVELET TRANSFORMATIONS*, ch. 5, pp. 183–229. John Wiley Sons, Ltd, 2019.
- [25] G. R. L. et al., “Pywavelets.” `PyWavelets/pywt: PyWavelets1.1.1`, 2019. Zenodo.
- [26] T. M. Mitchell, *Machine Learning*. USA: McGraw-Hill, Inc., 1 ed., 1997.
- [27] V. N. Vapnik, *The nature of statistical learning theory*. Springer-Verlag New York, Inc., 1995.
- [28] C. M. Bishop, *Pattern Recognition and Machine Learning (Information Science and Statistics)*. Berlin, Heidelberg: Springer-Verlag, 2006.
- [29] A. Géron, *Hands-on machine learning with Scikit-Learn and TensorFlow : concepts, tools, and techniques to build intelligent systems*. Sebastopol, CA: O’Reilly Media, 2017.
- [30] D. Kingma and J. Ba, “Adam: A method for stochastic optimization,” *Int. Conf. Learn. Representat.*, 12 2014.
- [31] S. Ruder, “An overview of gradient descent optimization algorithms,” *CoRR*, vol. abs/1609.04747, 2016.
- [32] “Espressif’s human-detection solution.” <https://youtu.be/tFxKUZEDSdw/>, accessed Apr. 24, 2021. Available on line.