Expandable Indoor Location Emulator For Machine Learning Models

Ricardo Alexandre[a], Rui Santos[b], Pedro Marques[b], Mário Antunes[a,b], João Paulo Barraca[a,b], João Silva[c], Nuno Ferreira[c]

[a] *Instituto de Telecomunicações, Universidade de Aveiro, Aveiro, Portugal*
[b] *DETI, Universidade de Aveiro, Aveiro, Portugal*
[c] *Think Digital, Aveiro, Portugal*

Abstract

This work describes an open-source indoor location emulator, based on a modular approach, with loosely coupled components, fitting multiple simultaneous location technologies, with a focus in the evaluation and training of Machine Learning methods. The work was implemented on top of the MQTT protocol, leading to a decoupled architecture that can easily be extended for several purposes, including integration of real and virtual devices. The emulator supports active and passive radio scenarios, including Wi-Fi and UHF RFID tags. The emulator is intended on underlying principles such as reusability and extensibility. The current implementation was used to evaluate an indoor location solution within a health environments with custom routes and different personnel.

Keywords: emulator, indoor location, rfid, rssi, web

Code Metadata

Nr.	Code metadata description	
C1	Current code version	v1.0.0
C2	Permanent link to code/repository used for this code version	`https://github.com/ATNoG/indoor-location-emulator`
C3	Code Ocean compute capsule	
C4	Legal Code License	MIT License
C5	Code versioning system used	git
C6	Software code languages, tools, and services used	HTML, CSS, JavaScript including some libraries (mapbox-gl.js, paho-mqtt.js, turf.js, map-gl-indoor), Python including multiple libraries for each module (see *requirements.txt* files on repository), Mosquitto MQTT broker, docker, docker-compose.
C7	Compilation requirements, operating environments & dependencies	Linux / Mac OS / Microsoft Windows with Windows Subsystem for Linux (WSL), virtual-environments, docker, docker-compose, modern web browsers that support WebGL. See `https://atnog.github.io/indoor-location-emulator/` for details.
C8	If available Link to developer documentation/manual	`https://atnog.github.io/indoor-location-emulator/`
C9	Support email for questions	rjfae@av.it.pt, mario.antunes@av.it.pt, jpbarraca@av.it.pt

Table 1: Code metadata

Software Metadata

Nr.	(Executable) software metadata description	
S1	Current software version	v1.0.0
S2	Permanent link to executable binaries of this version	`https://github.com/ATNoG/indoor-location-emulator`
S3	Permanent link to Reproducible Capsule	
S4	Legal Software License	MIT License
S5	Computing platforms/Operating Systems	Modern web browsers that support WebGL, docker, docker-compose.
S6	Installation requirements & dependencies	See `https://atnog.github.io/indoor-location-emulator/` for details.
S7	If available, link to user manual - if formally published include a reference to the publication in the reference list	See `https://atnog.github.io/indoor-location-emulator/` for details.
S8	Support email for questions	rjfae@av.it.pt, mario.antunes@av.it.pt, jpbarraca@ua.pt

Table 2: Software metadata

1. Motivation and significance

Information and Communications Technology (ICT) have undergone a natural evolution, aimed at improving people's quality of life. Currently, there has been a growing interest in systems capable of determining a user's location and providing services using this information [1, 2, 3, 4]. These systems are called Location Based Services (LBS) are characterised by the information or interface they present to the user is usually determined by the user's physical location. Generally integrates wireless and positioning technologies with location information management, providing consumers with the necessary services based on their geographical location, using Fingerprint location techniques. These techniques are divided into two phases, the offline Fingerprint calibration phase and the online location estimation phase [5]. In the offline phase, a set of Received Signal Strength Indicator (RSSI) values from various Access Points (APs) at each Reference Point (RP) is collected to form one of the Fingerprints stored in a database. During the online phase, the location of the terminal is obtained using pattern matching algorithms to compare with the database. The global spread and large increase of new LBS or context-aware computing, in which accurate target location information is required, have driven research in Indoor Positioning System (IPS), enable a range of location-based indoor tracking solutions, including Real-Time Location System (RTLS) in different application domains as museums and tourism [6, 7], gaming and augmented reality [8, 9], mall navigation [10], hospitals and monitoring of the elderly [11], train stations and airports [12], logistics or security, and emergency responders [13].

The Global Positioning System (GPS) is the most common technology used in LBS systems [14], was developed using trilateration techniques to locate a device in the world. However GPS and other systems based on trilateration or triangulation, which use time-of-arrival measurement perform poorly in indoor environment due to Non line-of-sight (NLOS) propagation effect [15].

There are several different technologies that can be used for indoor positioning, such as: Proximity-based systems, based on Radio-Frequency Identification (RFID) tags or beacons [16, 17, 18]; Wireless Local Area Network (WLAN)-based systems (Wi-Fi) [19]; Ultra-Wide-Band (UWB) [20, 21]; Bluetooth-based systems (Bluetooth or Bluetooth Low Energy (BLE)) [22, 23]; Ultrasounds-based systems (Acoustic) [24, 25]; Infrared (IR) [26, 27, 28]; Systems based on cellular networks (Global System for Mobile Communications (GSM), Code Division Multiple Access (CDMA), Long Term Evolution (LTE)) [29]. Whereas due to the complexity of internal layout, NLOS propagation environments, multipath effect and shadow fading [30, 31, 32], trilateral or triangulation methods, which are based on range or Time-of-arrival (TOA) angle, Time-difference-of-arrival (TDOA), Received Signal Strength (RSS) or Angle-of-arrival (AOA) measurements, suffer large location estimation errors. In fact, since WLAN techniques have been widely deployed in office premises and other hotspots, RSS values can be easily obtained from any Wi-Fi equipped terminal. The RSS-based Fingerprint location techniques in indoor WLAN environments have become relevant topic in research and scientific work [33, 34], in these last few years.

On the other hand, interest is also growing in the development of IPSs based on passive RFID technology. The main reason to choose RFID, in our case passive RFID is because passive tags support Backscattering, that is, they use the energy of the emitter signal to generate the response. So, it's not necessary a battery in each tag, as it happens in other technologies, this reduces substantially its costs (between approximately 0.05 - 0.10 €), focusing the investment in the infrastructure. This is a huge advantage over other technologies, since it eliminates the need for an entire battery management, the antennas can be disposed of in a more viable way. This feature makes this technology very suitable in a hospital or clinical setting, where there is a high density of medical equipment and medicines, and where each person can have their own RFID card or wristband.

This work as the goal of improving the localization of tags in indoor spaces, and considering infrastructures of the health sector, such as hospitals and clinical centres, where a large number of indicators for assessing the quality of services provided as well as the management of available resources are evaluated. It results in an open-source indoor location emulator, which implements modern web technologies for real time visualization, with a module-based approach of a decoupled architecture, using pluggable Machine Learning (ML) models, supporting RSSI standards for passive RFID scenarios. It takes into account the lack of similar software solutions capable of assembling, testing and validating ML solutions for this context, at a low cost, with loosely coupled patients. The Message Queuing Telemetry Transport (MQTT) protocol was selected as the communication method between emulator modules. It enable users to upload routes of indoor map locations, simulate patients / staff movements or equipment location to assemble, test and validate solutions to predict location accuracy with Machine Learning models. Also allow users to export the synthetic emulated data, for example to create prototypes of digital twins of indoor locations. The emulator is intended on underlying principles such as reusability and future extensibility.

The work presented is organized as follows: the software architecture of the indoor location emulator, detailing the architecture of the overall solution and of the individual modules, describing their functionalities and how it can be used in the evaluation of new indoor tracking solutions is presented in Section 2. Next, in Section 3, we present a characterization of the behavior observed in a real-world RFID environment. A use case/experimental scenario, the metrics to be used are described, the results and an analysis of them are presented in Section 4. Section 5 presents the reference examples. The rest sections contemplates the evaluated impact of the solution, in Section 6, the conclusions and future works, in Section 7 and finally the Conflict of Interest in carrying out this work, in Section 8.

2. Software Description

The indoor location emulator was developed, to solve a gap in existing open source solutions, which are vital for research in indoor location: how to validate indoor location data, train augmentation and prediction ML models, and visually inspect (or demonstrate) their accuracy with ease, over multiple technologies. When researching solutions for this area of research, it is not always clear which technologies are most suited. In the case of ML based solutions, it is also not always clear which features should be extracted and used. For some technologies (e.g. passive RFID), indirect features besides RSSI may present better datasets to construct models. Augmentation from multiple indoor location sources is also interesting to explore. Moreover, both for the purpose of evaluation and demonstration, a visual tool that integrates the real algorithms developed, presents a very strong use case for its development. Especially if it uses web technologies, building on the interaction capabilities of current browsers.

2.1. Software Architecture

The software is composed by a `frontend` component, a `backend`, a mqtt `broker`, and a number of `ml-models` calculating or predicting the location of moving assets (see Figure 1). The models are real software entities, which can be reused in real scenarios with real devices, or evaluated in a emulated environment. This allows researchers to refine existing algorithms, and avoids the need to re-implement algorithms in a simulator specific language, which significantly speeds up the development cycle.

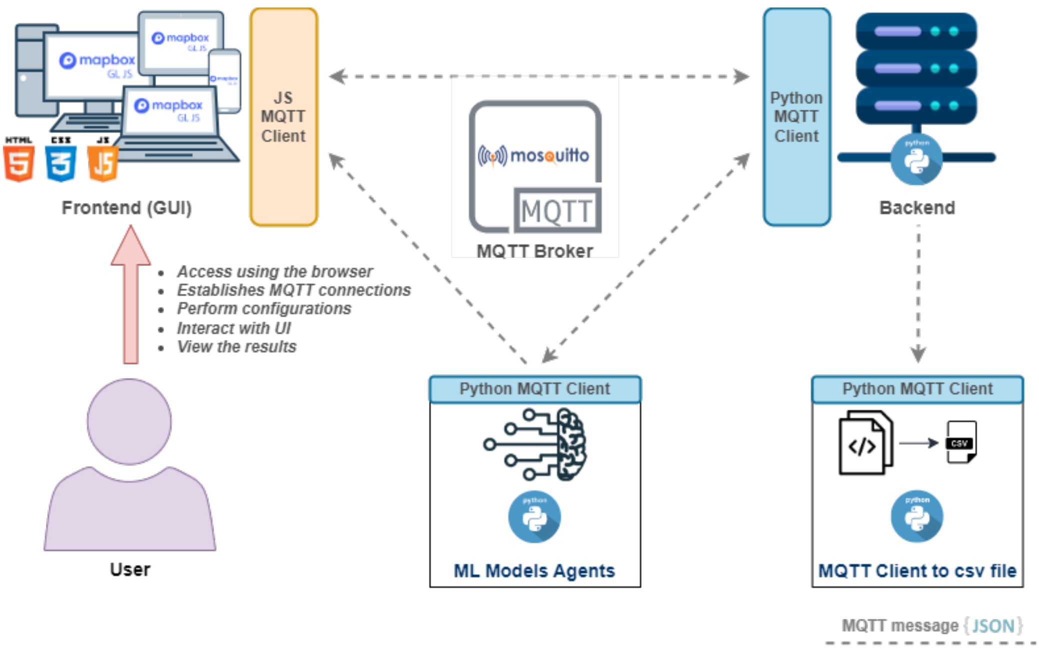

Figure 1: Schematic diagram of emulator elements.

Users interact directly with the `frontend` components. These provide a 2D visualization of indoor spaces of buildings, where the estimated positioning of an asset is presented in real time on a dynamic map. So as to maximize component reusability, and facilitate the evaluation of the location estimators, we opted by using standard geodesic coordinates (latitude and longitude). Moreover, we built upon existing web technologies (Mapbox GL JS [1]), to display a map and the assets, which resulted in a interface similar to other online map solutions running on a browser.

The `frontend` component, running on the browser uses JavaScript (JS) code (Vanilla JS) to implement its action logic, over an HyperText Markup Language (HTML)/Cascading Style Sheets (CSS) page. To further facilitate development, we opted for using real maps, in standard GeoJSON format as it would allow loading existing indoor maps, built directly or converted with tools such as QGIS [2]. An open source solution, map-gl-indoor [3], handles loading of indoor maps into GeoJSON files. Following standard naming conventions for GeoJSON maps, we can tag map elements with properties such as "level", and then walls, stairs or windows, resulting in a map with richer interaction. As an example, walls will be considered when calculated radio attenuation, and level information allow the emulator to consider more complex buildings with multiple levels.

The `frontend` implements all the interaction logic, providing the interface to users, and interacts with the `backend` components. These consist of a HyperText Transfer Protocol (HTTP) based server running a Python application, which is responsible for most of the data processing, mathematical calculations, feature augmentation and orchestration of the remaining components. Most importantly, it creates synthetic data from the current asset location, which is sent to location models. The result from the location models is converted into elements suited for display on the map interface.

Interacting with the `backend` component are the indoor location models. In our case, the purpose was to evaluate ML models, over an undetermined number of features, resulting in us naming these as `ml-models` implemented in Python.

The communication between the modules is ensured by the MQTT protocol over WebSockets (mosquitto) [4], and JavaScript Object Notation (JSON) messages, subscribing and publishing information through topics. This same approach is followed to interact with the location models, thus allowing a very high level of decoupling between the different components. If researchers

[1]https://mapbox.com
[2]https://qgis.org/
[3]https://github.com/map-gl-indoor/map-gl-indoor
[4]https://mosquitto.org/

which to develop aditional indoor location technologies, as long as they respect the message interface, any language, technology or approach can be used, without imposing changes to the remaining components. Actually, multiple location modules, in different languages can coexist in the same emulation session.

2.2. Software Functionalities

The main characteristics of `frontend` module consist of: The implementation of a MQTT client for communication (`paho-mqtt` library [5]); Load a building interior map, load the positions of Asset Points, load the positions of Antennas, load positions of Anchors (georeference points), load the positions of the Pulse Points (ML Algorithms results representation) making use of GeoJSON files; Display the lines of sight between Asset Points and Antennas as well as the intersection points with walls on each line of sight between Asset Points and Antennas; Animation of Asset Points with preloaded custom moves (GeoJSON file); Add and remove Antennas or Anchors on the map; Change the direction and opening angle of signal reception on Asset Points / tags; Move Asset Points, Antennas and Anchors on the map; Change the direction and opening angle of signal propagation of the Antennas, Anchors and Asset Points; Display the pulse points animated with the coordinates of the positions of the results of the various calculations (ML Algorithms).

The main characteristics of `backend` module consist of: The Implementation of an MQTT client for communication; Map walls capture; Distance calculation (between asset points and antennas); RSSI calculations (between asset points and antennas); Calculation of the intersections of the antennas signal propagation directions with the asset points sight lines; Implementation of an Orchestrator class (object that keeps all the information about the system); Establishes the communication with the `frontend` and ML Agents through the MQTT protocol using JSON messages.

An independent module called `static-files` includes all the static-files used in `frontend` and `backend` modules, such as antennas datasets, configuration files, images, MQTT broker configuration files. Highlighting a set of scripts of writing messages to files developed in Python language using several tools and open source libraries. This is responsible for storing the data from the calculations performed in the `backend`, that will be used by the ML Agents, in the process of training and learning and validation of the ML models. It was included as an utility script (`static-files/utils` module

[5] https://www.eclipse.org/paho/index.php?page=clients/python/index.php

directory) and presents the main characteristics: the implementation of an MQTT client for communication; Write the RSSI or number of activations values calculated for all the antennas and the Asset Point position coordinates (Longitude, Latitude) received, in .Comma-Separated Values (CSV) format files. This is also the module where we can add different locations to emulate results, by creating a folder with the name of the new location to be included and adding its configuration files (GeoJSON files).

3. Modelling of real world RFID behavior

RSSI is a measure of the strength of a signal received by a device. An RSSI-based localization algorithm follows the principle that a signal emitted by a device gradually loses power as the distance at which it is propagated increases, until it reaches a distance at which the signal naturally fades away. The utilization of the RSSI data from several devices (converted to distance from the transmitter) can be used to calculate the position through trilateration or fingerprinting techniques. The correspondence that exists between power and distance is the basis for establishing the localization. The decay, in the overwhelming majority of cases, is not linear and much less uniform, varying with the direction of propagation due to obstacle reflections or interference. The signal propagation itself is often influenced by external factors such as temperature and humidity, which prevents it from being considered constant over relatively large areas. Thus, the application of this type of system implies a high measurement and modeling effort to the environment in which they are implemented, which limits their use. The accuracy of the system depends largely on the degree of extensiveness of the adaptation performed, as well as the signal transmission power. The relationship between RSSI values and distance is represented by the path loss model, defined by equation Equation 1:

$$PL = P_{Tx_{dBm}} - P_{Rx_{dBm}} = PL_0 + 10 \times \gamma \times log10(\frac{d}{d_0}) + Xg \tag{1}$$

Where:

- PL is the total path loss in decibels (dB).
- $P_{Tx} = 10 \times log10(\frac{P_{Tx}}{1mW})$ is the transmitted power in dBm where P_{Tx} is the transmitted power in watts.
- $P_{Rx} = 10 \times log10(\frac{P_{Rx}}{1mW})$ is the received power in dBm where P_{Rx} is the received power in watts.
- PL_0 is the path loss in decibels (dB) at the reference distance calculated using Friis' free-space path loss model.
- d is the path length.

- d_0 is the reference distance, usually 1 m (or 1 mile) for a large cell and 1 m to 10 m for a microcell.
- γ (or n) is the path loss exponent.
- Xg (or σ) is a normal (or Gaussian) random variable with zero mean, reflecting the attenuation (dB) caused by fading.

From this association results the equation to calculate the RSSI, defined in equation Equation 2.

$$RSSI = RSSI_0 - 10 \times n \times log10(\frac{d}{d_0}) + Xg \tag{2}$$

Where:
- $RSSI_0$ is the signal strength measured in dBm, at distance d_0.
- **Multipath constant (n):** Multipath or Path Loss Exponent causes the signal to arrive in different ways at the device due to reflections, diffraction and scattering, depending on the internal distribution such as furniture, walls, objects, etc.
- **Fading constant (Xg):** Fading is the time variation of the received signal. It depends on the environment and the movement of the devices.

For the usability tests of the tool, on the RSSI Path Loss model was applied the empirical coefficient values for indoor propagation defined in equation Equation 2, based on an "Office With soft partition" (Frequency of Transmission = 900MHz):
- $RSSI_0$ is the signal strength, at distance d_0, estimated in -20 dBm.
- d is the path length between an antenna and an asset point calculated by the `backend` module of the emulator;
- d_0 is the reference distance, in this case for a microcell is 1 $meter$;
- n (or γ) is the path loss exponent, with coefficient estimated in 2.4;
- Xg (or σ) is a normal (or Gaussian) random variable with zero mean, reflecting the attenuation (dB) caused by fading, with coefficient estimated in 9.6.

The module was adapted to also support another type of data sent by the emulator. Instead of using the RSSI data generated by the known number of antennas located at the various points of the building to generate the coordinates and return them to the `frontend` module, the "Predictor" uses the data related to the number of activations of each tag. This feature was selected, as it demonstrated a good capability to track tag distance to the antenna, or at least at par with RSSI. In this way, a set of experimental data was collected from real hardware consists of an Antenna RF and RFID passive tags from 0.5 meters until 10.0 meters with steps of 0.5 meters, applying power scans at 280, 290, and 300 mW. The dataset was included as a look up table data that correlate the number of activations with the distance between an Asset Point and an Antenna RF, emulated in system. The emulator will estimate the number of activations in the `backend` module made use of the features explicit in configuration file.

The models developed which use the number of activations recorded on an antenna to estimate its distance from the asset. Since the `Predictor` is responsible for returning the geographic coordinates (2 dimensions, Latitude and Longitude) of the asset, it was necessary to develop a method to translate the distance relative to an antenna (1 dimension) returned by the `ml-models` module. This method defines the asset point coordinates based on the antenna's coordinates, direction line and the distance calculated by the algorithms (see Equation 3 and Equation 4).

$$asset_{longitude} = antenna_{longitude} - \sin(antenna_{angle}) \times distance \tag{3}$$

$$asset_{latitude} = antenna_{latitude} + \cos(antenna_{angle}) \times distance \tag{4}$$

Where:

- $asset_{longitude}$, $asset_{latitude}$ and $antenna_{longitude}$, $antenna_{latitude}$ represent the longitude and latitude coordinates (in meters) of the asset and antenna points respectively.
- $antenna_{angle}$ is the angle of the antenna's direction (North = 0°) and distance is the distance between the asset and the antenna.

Due to certain limitations of the tested hardware (antenna RF and passive RFID tags), it was defined that values higher than 10 metres would not be calculated.

4. Use Case and Results

In order to create synthetic data concerning the number of tag activation events, an interpolation was made based on the results obtained in experimental data acquisition procedure from 0 meters until 10 meters, in which anchors were used and activation numbers were recorded at a greater distance (up to 9 m) than in the other experiments. The interpolation was applied to each value of the 3 selected powers (280, 290 and 300 mW) and was based on the reverse sigmoid function, which fits the behaviour observed in the experimental data acquisition procedure, and was set to 0 after 10 m as it was observed in the experiment. The One-Class SVM outlier detect algorithm was also applied for outlier detection and removal using the Scikit-learn library [6]. A method of generating synthetic noise from the standard deviation recorded in same experiment was also added, resulting in more credible and realistic data. An interpolation result based on the reverse sigmoid function was applied on the number of activation events recorded in the experimental data acquisition procedure.

These initial results where explored further in the following work [35].

5. Illustrative Examples

The Figure 2 present the emulator Graphical User Interface (GUI), after the connection with MQTT Broker is established, show all the selected ML models, including the graphic and textual visualization of the positioning calculations. One Asset Point, ten Antennas and all the five ML models results are disposed in the map.

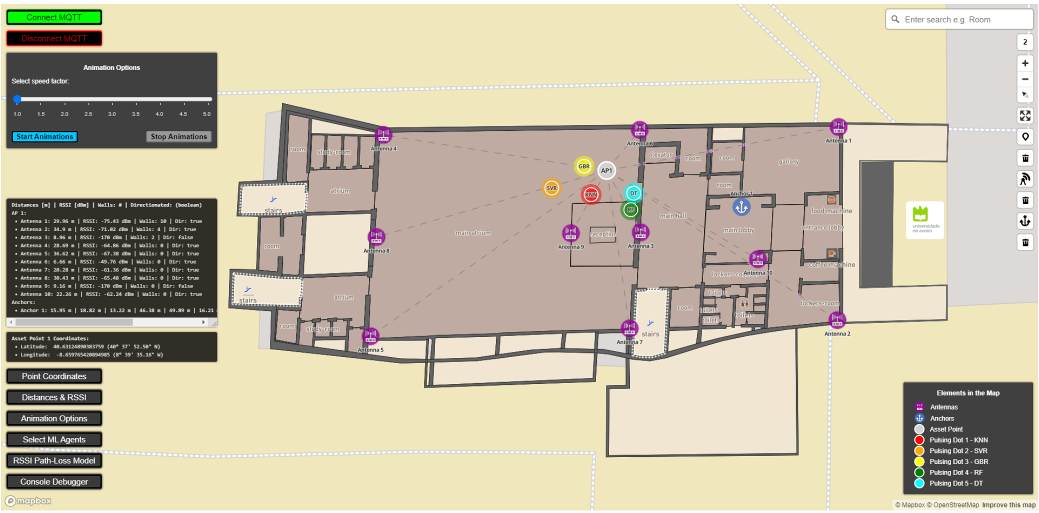

Figure 2: Emulator GUI, accessed on Computer browser, presenting the moment after the connection is established, all the selected ML models, including the graphic and textual visualization of the positioning calculations.

The map presented in the emulator is a 2D representation of the 2nd floor of University of Aveiro Library, and it was obtained through the drawing of the official plan of the building. This space is presented as a base example for

[6]https://scikit-learn.org/stable/modules/generated/sklearn.svm.OneClassSVM.html

validation of the developed tool. It was chosen because it is a public space whose floor plan is of general access to the community. Other spaces can be included in the emulator, through previous configuration by including the files of a new place previously created and updating the execution code of the tool. Because we use open formats (GeoJSON), any custom map can be added to the tool. Actually, even outdoor maps can be used, although that was not extensively explored by the authors.

Following we describe the graphical emulator elements that allow the user to interact with the developed tool:

- **In the upper left corner** there are the connect and disconnect buttons to the MQTT broker. The connect button opens an MQTT connection data parameterization panel. The disconnect button only disconnects the session once established.
- **In the upper right corner** there is the tools zone, including search, navigation, zoom, rotation, add and remove asset points, add and remove antennas, add and remove reference points (anchors).
- **In the lower left corner** there is a set of buttons for interaction with the session, including one to show Asset Points coordinates, another to show Distances and RSSI values calculated for the Antennas where two panels are displayed with that information. In the case of the Antennas information, this is only displayed after the connection is established. It also includes a button to open and close the animation options panel of the Asset Point, another to display the ML agent selection panel, another to open the parameterization panel of the data needed for the RSSI calculations using the path-loss model, and finally a button to open and close a debug console, with all the printed session information.
- **In the lower right corner** a static legend of the active map elements is displayed.

6. Impact

The Expandable Indoor Location Emulator For Machine Learning Models demonstrates a consistent approach, that couples multiple services together, and simplifies the process of acquiring synthetic data of RSSI measurements using RFID tags, based on a real world approach for an indoor location environments. The data model and data flow as well as the tool itself can be easily extended to include another set of machine learning data models or interactions with new services.

The integration, made through a message broker, allows mixed scenarios, with real devices or software components, interacting together on the same environment. This facilitates training of models with larger datasets, and the integration and development of components, every time that access to real devices is somewhat limited on constrained.

Having a web visualization frontend will also facilitates the development of ML solutions as it allows researchers to both view and control assets in real time. It also becomes possible to assess the quality of the ML estimators directly on map, which is extremely useful in early stages of model development.

The tool was implemented as part of the EU funded research project "SDRT Health" solution which consists of the development of a new, more advanced antenna system, which will be supported on an Software Defined Network (SDN) system with Artificial Intelligence (AI) that will reduce the overall Radio-Frequency (RF) emission with smaller antennas and with less impact, applied in a context of management of health systems, seeking to improve resource management performance indicators.

7. Conclusions

In this paper, the Expandable Indoor Location Emulator For ML Models has been described - an open-source web tool for the application, study, and evaluation of a set of real characteristic factors, that help to assemble, test and validate solutions to predict location accuracy provided by Machine Learning models to RSSI standards for passive RFID scenarios, applied to a context of improving performance indicators of health management systems, to serve hospitals and clinic centers. It enable users to upload routes of indoor map locations, simulate patients / staff movements. The module-based approach has been applied to describe the data model in emulator and automate the process of synthetic data acquisition data. Furthermore, a set of tests has been employed to validate the tool.

8. Conflict of Interest

We wish to confirm that there are no known conflicts of interest associated with this publication and there has been no significant financial support for this work that could have influenced its outcome.

Acknowledgment

This work is supported by the European Regional Development Fund (FEDER), through the Competitiveness and Internationalization Operational

Programme (COMPETE 2020) of the Portugal 2020 framework [Project SDRT with Nr. 070192 (POCI-01-0247-FEDER-070192)].

References

[1] J. A. Gómez, A. V. Medina, S. Martín, E. Dorronzoro, and O. Rivera, "Fingerprint indoor location simulator for aal," *Journal of Ambient Intelligence and Smart Environments*, pp. 109–124, 2016.

[2] G. Félix, M. Siller, and E. N. Álvarez, "A fingerprinting indoor localization algorithm based deep learning," in *8th International Conference on Ubiquitous and Future Networks*, 2016, pp. 1006–1011.

[3] M. C. Pérez-Rubio, A. Hernández, D. Gualda-Gómez, S. Murano, J. Vicente-Ranera, F. Ciudad-Fernández, J. M. Villadangos, and R. Nieto, "Simulation Tool and Online Demonstrator for CDMA-Based Ultrasonic Indoor Localization Systems," *Sensors*, vol. 22, no. 3, 2022.

[4] B. El Boudani, L. Kanaris, A. Kokkinis, M. Kyriacou, C. Chrysoulas, S. Stavrou, and T. Dagiuklas, "Implementing Deep Learning Techniques in 5G IoT Networks for 3D Indoor Positioning: DELTA (DeEp Learning-Based Co-operaTive Architecture)," *Sensors*, vol. 20, no. 19, 2020.

[5] G. Ding, J. Zhang, Z. Tan *et al.*, "Overview of received signal strength based fingerprinting localization in indoor wireless LAN environments," in *5th IEEE Int. Symp. on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, 2013, pp. 160–164.

[6] M. Fleck, M. Frid, T. Kindberg, E. O'Brien-Strain, R. Rajani, and M. Spasojevic, "From informing to remembering: Ubiquitous systems in interactive museums," *IEEE Perv. Comp.*, vol. 1, no. 2, pp. 13–21, 2002.

[7] P. Spachos and K. N. Plataniotis, "Ble beacons for indoor positioning at an interactive iot-based smart museum," *IEEE Systems Journal*, vol. 14, no. 3, pp. 3483–3493, 2020.

[8] Z. W. Ong, "Indoor navigation with augmented reality," Ph.D. dissertation, Tunku Abdul Rahman University College, 2021.

[9] R. Joshi, A. Hiwale, S. Birajdar, and R. Gound, "Indoor navigation with augmented reality," in *ICCCE 2019*. Springer, 2020, pp. 159–165.

[10] V. Renaudin, M. Ortiz, J. Perul, J. Torres-Sospedra, A. R. Jiménez, A. Pérez-Navarro, G. M. Mendoza-Silva, F. Seco, Y. Landau, R. Marbel *et al.*, "Evaluating indoor positioning systems in a shopping mall: The lessons learned from the ipin 2018 competition," *IEEE Access*, vol. 7, pp. 148 594–148 628, 2019.

[11] R. Montoliu, E. Sansano, A. Gascó, O. Belmonte, and A. Caballer, "Indoor positioning for monitoring older adults at home: Wi-fi and BLE technologies in real scenarios," *Electronics*, vol. 9, no. 5, p. 728, 2020.

[12] B. Molina, E. Olivares, C. E. Palau, and M. Esteve, "A multimodal fingerprint-based indoor positioning system for airports," *Ieee Access*, vol. 6, pp. 10 092–10 106, 2018.

[13] C. Fischer and H. Gellersen, "Location and navigation support for emergency responders: A survey," *IEEE Pervasive Computing*, vol. 9, no. 01, pp. 38–47, 2010.

[14] L. Jaulin, "5 - instantaneous localization," in *Mobile Robotics*, L. Jaulin, Ed. Elsevier, 2015, pp. 171–196.

[15] Y. Chen, D. Lymberopoulos, J. Liu, and B. Priyantha, "Indoor localization using fm signals," *IEEE Transactions on Mobile Computing*, vol. 12, no. 8, pp. 1502–1517, 2013.

[16] K. Lorincz and M. Welsh, "A robust, decentralized approach to rf-based location tracking," *Harvard University, Cambridge, MA, Tech. Rep. TR-19-04*, 2004.

[17] P. Barsocchi, F. Furfari, P. Nepa, and F. Potortì, "Rssi localisation with sensors placed on the user," in *2010 International Conference on Indoor Positioning and Indoor Navigation*. IEEE, 2010, pp. 1–6.

[18] S. Hossain, S. H. Ariffin, N. Fisal, C. K. Neng, N. A. Hassan, and L. Latiff, "Accuracy enhancement of fingerprint indoor positioning system," in *3rd International Conference on Intelligent Systems Modelling and Simulation*. IEEE, 2012, pp. 600–605.

[19] L.-w. Chan, J.-r. Chiang, Y.-c. Chen, C.-n. Ke, J. Hsu, and H.-h. Chu, "Collaborative localization: Enhancing wifi-based position estimation with neighborhood links in clusters," in *International Conference on Pervasive Computing*. Springer, 2006, pp. 50–66.

[20] S.-G. Kwon, O.-J. Kwon, K.-R. Kwon, and S.-H. Lee, "UWB and MEMS IMU Integrated Positioning Algorithm for a Work-Tool Tracking System," *Applied Sciences*, vol. 11, no. 19, 2021.

[21] A. Alarifi, A. Al-Salman, M. Alsaleh, A. Alnafessah, S. Al-Hadhrami, M. A. Al-Ammar, and H. S. Al-Khalifa, "Ultra wideband indoor positioning technologies: Analysis and recent advances," *Sensors*, vol. 16, no. 5, 2016.

[22] R. Bruno and F. Delmastro, "Design and analysis of a bluetooth-based indoor localization system," in *Personal Wireless Communications*, M. Conti, S. Giordano, E. Gregori, and S. Olariu, Eds. Berlin, Heidelberg: Springer Berlin Heidelberg, 2003, pp. 711–725.

[23] F. Pascale, E. A. Adinolfi, M. Avagliano, V. Giannella, and A. Salas, "A Low Energy IoT Application Using Beacon for Indoor Localization," *Applied Sciences*, vol. 11, no. 11, 2021.

[24] M. Hazas and A. Ward, "A novel broadband ultrasonic location system," in *Int. Conf. on Ubiquitous Computing*. Springer, 2002, pp. 264–280.

[25] N. B. Priyantha, A. K. Miu, H. Balakrishnan, and S. Teller, "The cricket compass for context-aware mobile applications," in *7th Annual Int. Conf. on Mobile Computing and Networking*, 2001, pp. 1–14.

[26] R. Azuma, "Tracking requirements for augmented reality," *Communications of the ACM*, vol. 36, no. 7, pp. 50–51, 1993.

[27] A. Harter and A. Hopper, "A distributed location system for the active office," *IEEE Network*, vol. 8, no. 1, pp. 62–70, 1994.

[28] R. Want, A. Hopper, V. Falcao, and J. Gibbons, "The active badge location system," *ACM Trans. on Inf. Syst.*, vol. 4, pp. 42–47, 1997.

[29] S. Tekinay, "Wireless geolocation systems and services," *IEEE Communications Magazine*, vol. 36, no. 4, pp. 28–28, 1998.

[30] F. Zafari, A. Gkelias, and K. K. Leung, "A survey of indoor localization systems and technologies," *IEEE Communications Surveys & Tutorials*, vol. 21, no. 3, pp. 2568–2599, 2019.

[31] C. McElroy, D. Neirynck, and M. McLaughlin, "Comparison of wireless clock synchronization algorithms for indoor location systems," in *IEEE International Conference on Communications Workshops*, 2014, pp. 157–162.

[32] A. Boukerche, L. A. Villas, D. L. Guidoni, G. Maia, F. D. Cunha, J. Ueyama, and A. A. Loureiro, “A new solution for the time-space localization problem in wireless sensor network using UAV,” in *3rd ACM Int. Symp. on Design and Analysis of Intelligent Vehicular Networks and Applications*, 2013, pp. 153–160.

[33] B. N. Ozdemir and A. Ceylan, “Constructing a precise radio map and application of indoor positioning with dual-frequency Wi-Fi fingerprinting method,” *Measurement*, vol. 163, p. 107997, 2020.

[34] G. Ding, J. Zhang, l. Zhang, and Z. Tan, “Overview of received signal strength based fingerprinting localization in indoor wireless LAN environments,” *5th IEEE Int. Symp. on Microwave, Antenna, Propagation and EMC Technologies for Wireless Communications*, 2013.

[35] R. Santos., R. Alexandre., P. Marques., M. Antunes., J. Paulo Barraca., J. Silva., and N. Ferreira., “Towards improved indoor location with unmodified rfid systems,” in *Proceedings of the 12th International Conference on Pattern Recognition Applications and Methods - ICPRAM*, INSTICC. SciTePress, 2023, pp. 156–163.