

Article

Enhancing Indoor Positioning Accuracy with WLAN and WSN: A QPSO Hybrid Algorithm with Surface Tessellation

Edgar Scavino ^{1,*}, Mohd Amiruddin Abd Rahman ¹, Zahid Farid ², Sadique Ahmad ³
and Muhammad Asim ^{3,*}

¹ Department of Physics, Faculty of Science, Universiti Putra Malaysia (UPM), Serdang 43400, Malaysia; mohdamir@upm.edu.my

² Department of Electrical Engineering, Abasyn University, Peshawar 25000, Pakistan; zahid.farid@abasyn.edu.pk

³ EIAS Data Science Lab, College of Computer and Information Sciences, Prince Sultan University, Riyadh 11586, Saudi Arabia; ahmad01.shah@ieee.org

* Correspondence: edgarscavino@upm.edu.my (E.S.); masim@psu.edu.sa (M.A.)

Abstract: In large indoor environments, accurate positioning and tracking of people and autonomous equipment have become essential requirements. The application of increasingly automated moving transportation units in large indoor spaces demands a precise knowledge of their positions, for both efficiency and safety reasons. Moreover, satellite-based Global Positioning System (GPS) signals are likely to be unusable in deep indoor spaces, and technologies like WiFi and Bluetooth are susceptible to signal noise and fading effects. For these reasons, a hybrid approach that employs at least two different signal typologies proved to be more effective, resilient, robust, and accurate in determining localization in indoor environments. This paper proposes an improved hybrid technique that implements fingerprinting-based indoor positioning using Received Signal Strength (RSS) information from available Wireless Local Area Network (WLAN) access points and Wireless Sensor Network (WSN) technology. Six signals were recorded on a regular grid of anchor points covering the research surface. For optimization purposes, appropriate raw signal weighing was applied in accordance with previous research on the same data. The novel approach in this work consisted of performing a virtual tessellation of the considered indoor surface with a regular set of tiles encompassing the whole area. The optimization process was focused on varying the size of the tiles as well as their relative position concerning the signal acquisition grid, with the goal of minimizing the average distance error based on tile identification accuracy. The optimization process was conducted using a standard Quantum Particle Swarm Optimization (QPSO), while the position error estimate for each tile configuration was performed using a 3-layer Multilayer Perceptron (MLP) neural network. These experimental results showed a 16% reduction in the positioning error when a suitable tile configuration was calculated in the optimization process. Our final achieved value of 0.611 m of location incertitude shows a sensible improvement compared to our previous results.

Keywords: quantum particle swarm optimization; indoor localization; fingerprinting; WiFi; wireless sensor network; multilayer perceptron; surface tessellation



Citation: Scavino, E.; Abd Rahman, M.A.; Farid, Z.; Ahmad, S.; Asim, M. Enhancing Indoor Positioning Accuracy with WLAN and WSN: A QPSO Hybrid Algorithm with Surface Tessellation. *Algorithms* **2024**, *17*, 326. <https://doi.org/10.3390/a17080326>

Academic Editor: Mohammad Majid al-Rifaie

Received: 5 April 2024

Revised: 18 July 2024

Accepted: 18 July 2024

Published: 25 July 2024



Copyright: © 2024 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

1. Introduction

In recent years, the fast development of wireless technologies and their ubiquitous applications [1–3] have drawn a lot of attention for both outdoor and indoor localization tracking implementations [4–8]. As far as outdoor applications are concerned, the most known and popular technology is the satellite-based Global Position System (GPS), which offers uncountable services in the domains of tourism, navigation, military, etc. [9]. However, because of the satellite origin of the GPS signals, this technology loses performance in urban canyons, near tall walls, and in bad weather due to the weakness of the received

signals, becoming completely useless in many indoor and underground environments. Consequently, GPS becomes an unreliable and ineffective method in scenarios where precise indoor localization is required [10].

In order to overcome these limitations, wireless technologies are being widely implemented for indoor positioning purposes. These techniques include Radio Frequency Identification (RFID) [11,12], and Wireless Local Area Networks (WLANs) [13–15], Bluetooth [16], Wireless Sensor Networks (WSNs), among others [17–19]. Other methods, based on the angle-of-arrival and the time-of-arrival, depend on measurements that require additional hardware and are therefore less common [20,21].

Thanks to the diffusion of smartphones and the availability of free WiFi in most indoor public spaces, the implementation of many of these communication technologies can be easily and inexpensively undertaken. As a consequence, location-based services are rapidly expanding with several benefits, mainly driven by the common availability of the appropriate hardware, both in terms of signal generation and reception.

The most common approach to infer the position of a receiver in a place where many wireless signals are available is based on the Received Signal Strength (RSS), which can be measured without the need to decode any communication feature. Localization is consequently determined by the pattern matching of the intensity of multiple WLAN sources [22]. Nevertheless, even the hybrid technology of a WiFi and WLAN combined approach for indoor positioning algorithms can lead to unavoidable uncertainty, mainly driven by signal instability and fading, reflection on structures, the presence of people, and possibly autonomous trolleys [23,24]. In this work, the localization approach was based on the fingerprinting matching process from both WLAN and WSN signals as the basic scheme of positioning determination. The two technologies complemented each other in terms of spatial behavior, as WiFi signals penetrate walls, while WLAN presents a shorter range with a stronger gradient [25].

While a pattern-matching approach is classified as a deterministic algorithm [26], the spatial distributions of the signal intensities are strongly non-linear in the whole research space and thus require the implementation of methods such as neural networks and heuristic optimization techniques. Consequently, an Artificial Neural Network (ANN) approach guarantees robustness against noise and interference, which are among the major factors affecting the accuracy of indoor positioning systems.

In this work, we extended the concept of surface tessellation, in which we subdivided the research area into identical, discrete rectangular tiles. This approach was based on the idea that most of the time, a coarse location precision was sufficient to infer the actual position of a receiver and its carrier. In our previous work on research area tessellation [27], square tiles of 1 m side were considered as proof of concept. The published results, together with the improvements obtained by signal weighing and an appropriate ANN architecture, encouraged us to further investigate the tessellation technique for indoor positioning.

The core contributions of this paper are listed as follows:

1. To extend previously published results on the same dataset by joining our signal optimization results and the concept of tessellation of the search space.
2. To develop an optimization algorithm by investigating a suitable tessellation of the research space to reduce the overall error in the Multilayer Perceptron positioning algorithm.
3. To introduce a heuristic approach for the determination of regular space meshes in general positioning algorithms. With this fact in mind, a Quantum Particle Swarm Optimization (QPSO) algorithm was integrated into the positioning neural network application.

The remaining sections of the paper are organized as: Section 2 gives a brief description of the related work, including our previous analysis of the studied dataset. Section 3 presents in more detail the approach and techniques we used in this research. Here, particular attention is given to the QPSO process and to the improvement of the MLP application in space tessellation, which was used in our previously published works. The following part of this paper presents the results of this study, with references to the optimization of the MLP and the performance of our proposed QPSO algorithm. The

achievements of our optimization process are presented and compared with results both from the literature and from the authors' previous research. Finally, the conclusions of this work are briefly discussed, and projects and ideas for future research are introduced.

2. Related Work

WiFi technology has been extensively researched for both communication and indoor positioning systems. However, relying solely on WiFi can result in poor performance due to signal fading and distortions caused by infrastructure and human presence. To address these issues, additional readily available sources of wireless signals, such as magnetic fields, inertial measurement units, and wireless sensor networks (WSN), have been incorporated.

Chen et al. [28] integrated multiple sensors, including an Inertial Measurement Unit (IMU), a magnetometer, and a WLAN RSSI (RSS Indicator). They proposed a Euclidean matching algorithm to estimate the location based on the WiFi signal. The Robust Extended Kalman Filter algorithm was used to estimate the location based on the IMU sensor. Additionally, an enhanced dynamic time-warping algorithm is used to match the magnetic fingerprint location based on magnetometer measurement. Although multiple sensor systems improved the accuracy of the positioning system, the computation required user mobile processing power to obtain the location information, which might be unfeasible for real-time applications.

Luo et al. [29] proposed a joint WSN–WiFi approach for improving indoor positioning estimation. The work considered the communication between sensor nodes as well as between the sensor nodes and the receiver to help with user positioning. The detection algorithm considered a grid-based search, with the computation cost increasing as the search location was expanded due to the increment in the number of grids to be considered. Khan et al. [27] also proposed a WSN–WiFi technique based on a grid search that considered the floor tile spacing. This technique estimates the location of the user to be within the respective boundaries of the specific tile's location based on an appropriately trained neural network.

Guo et al. [30] included WiFi and cellular telephony network RSS signals in a hybrid approach for an indoor positioning algorithm. The fingerprinting database was analyzed by Principal Component Analysis (PCA) with interpolations of the raw data. Subsequently, data were processed by an ANN separately as cellular signals were first used to provide coarse training, which was then improved using the WiFi data. This hybrid approach improved the positioning precision; however, the process was very demanding in terms of computing time and unsuitable for real-time applications.

The synergy between multiple smartphone sensors was investigated by Gang and Pyun [31], who joined the signals of Bluetooth, magnetic field sensors, IMU, and a camera together. Their accomplishments demonstrated the effectiveness of a hybrid approach, in which combined heterogeneous sources of signals provided better indoor positioning results. However, their method required a permanent connection to a central server, and they chose not to use WiFi signals, which were deemed to be variable and unreliable in RSS fingerprinting algorithms.

This work implements a consolidated hybrid approach as a positioning system by means of combined RSS signals from WiFi and WSNs in order to improve localization accuracy in indoor spaces. The utilization of WiFi–WSN signals does not require any additional computation in end-user devices and can be used for real-time applications. In this work, we apply the results of our previous research on relative signal weighing and MLP architecture optimization, which led to an overall significant reduction of the positioning error on the same dataset [32]. We also utilize a similar algorithm based on standard QPSO, which we integrate with the MLP neural network in order to improve the accuracy of the localization system. The detailed method, which aims to find values of appropriate tile sizes and tessellation corners, is introduced in the next section.

3. Research Methodology

Due to their widespread deployment and access point availability, WLAN and WSN-based technologies are gaining interest in the domain of indoor localization and positioning services. RSS data from WLAN and WSN sources are becoming increasingly important for refining and improving the accuracy in indoor environments [13,17], particularly when GPS signals become unreliable or undetectable. In this work, our goal consists of the implementation of a refinement of previous research, again by using a hybrid indoor positioning system in which both signals from available WLAN access points and WSN-based technologies are considered.

In particular, the possibility of further refinement and optimization is based on two previous distinct works, in which we separately considered the possibility of weighting the WLAN and WSN source signals and the tessellation of the research space into a regular grid of square tiles. A preliminary inspection of the collected data confirmed the lack of linearity of signal intensities with respect to the relative positions inside the research area. This aspect reinforced the need to process the data by means of a machine learning approach, in which we included the approach of weighting the available signals from both WLAN and WSN with the aim of improving indoor positioning accuracy.

The concept of tessellation of the research space is based on the fact that, for many applications, the precise localization of a signal receiver and its end-user is not required. Furthermore, any positioning algorithm presented and discussed in the literature carries an uncertainty of the order of magnitude of 1 m. As a result, for most indoor positioning applications, finite-sized tiling would not become a limiting factor in precision for localization purposes, especially in large indoor environments. For this reason, this work focuses on the optimization of a regular tessellation of the research space, seeking optimal sizes of rectangular adjacent tiles and the collocation of the corner of the resulting grid.

3.1. Signal Acquisition

In order to address the subject of an accurate indoor positioning system, raw data collection, and preprocessing are significant steps for location fingerprinting. This study investigates a collection of data that were measured as RSSI values based on an IEEE 802.11b/g wireless card. RSSI data were acquired at the main basement level of the Faculty of Engineering and Built Environment at the Universiti Kebangsaan Malaysia (UKM) [33]. In January 2014, data were collected on a single day without interruption. While the research area was regular in shape and did not present any obstacles, the boundaries were quite heterogeneous, with research laboratories, furnished offices, lecture halls, structural pillars, and people occasionally crossing the place. The research area and the schedule of the data acquisition were purposely chosen in order to reproduce the normal conditions of a typical working day. This way, disturbances in the signals were taken into account as they were received and measured.

Figure 1 shows a sketch of the research area and its irregular boundaries. A regular grid including 96 anchor points encompasses the unobstructed part, in which WLAN and WSN signals were recorded regardless of people and materials transiting through the experimental area. For experimental RSS data collection, we selected a regular grid of 96 anchor points for the creation of the fingerprinting database, in an array of 6×16 points, as a consequence of the chosen horizontal and vertical separation between anchor points, respectively, 1 m and 1.5 m in the directions parallel to the main walls. The height of each anchor point was strictly kept at the value of 1 m in order to avoid variations in the locally measured signal intensities and to remove the height as a possible sensitive variable [34].

A Lenovo G580 laptop with a core i5 processor and Windows 7 operating system was used as a measurement setup. For the collection of RSS data samples, an Atheros AR9285 onboard wireless adaptor based on 802.11n was added. An “inSSIDer” open WiFi Scanner software (version 4.2) was used, to which some adaptations were executed in order to create the log files of the measured RSS signals with respect to the visible MAC address and the time stamp.

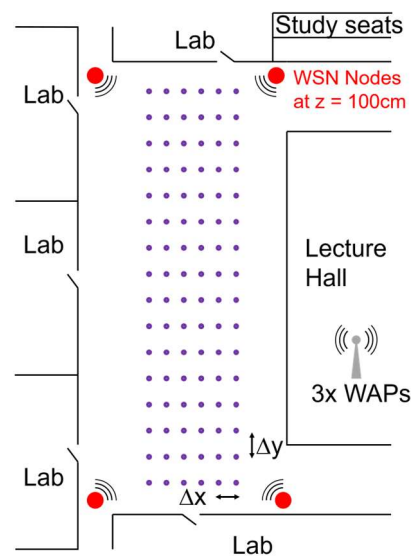


Figure 1. A sketch of the research area, with depictions of irregular boundaries, signal sources, and the grid of anchor points where signal intensities were recorded.

The signal intensities in dB from three available WiFi and four installed WSN signal sources were recorded simultaneously 300 times for one minute, i.e., every 0.2 s, in each of the 96 anchor points. Only three signals from the WSN were considered for the database, as one of the access points was faulty and provided random, unreliable values. The average of each set of 300 values was used to build the signal intensity database. The stability of all six available signals was checked through the standard deviation of each set, which was satisfactorily within 5% of the average value for all anchor points. The acquired data were organized and saved in a matrix of 96×8 elements. The WiFi data were stored in columns 1–3; the WSN data were placed in columns 4–6; and columns 7 and 8 contained the spatial coordinates of the 96 anchor points. In order to utilize the appropriate signal weighting, which provided a significant reduction in the positioning error in our previous work [32], it was strictly necessary to identify and keep the same signal sequence in all matrix columns. At the same time, the meaning of this research is fundamentally dependent on the constant positions of all access points and of any other signal source.

3.2. Data Analysis

The procedural scheme of the proposed optimization algorithm is based on a number of fundamental sections. First, data from hybrid sources were acquired and stored using fingerprinting and RSS principles. Data were kept as they were recorded, without any particular preprocessing, any further analysis, or suggestion of a limited database.

The second step consisted of creating a mathematical model for both the distribution of the signal data and a mainframe for the tessellation of the research space. As the behavior of the measured signal intensities along the 2D space of the experimental area proved to be inherently non-linear, a Multilayer Perceptron (MLP) neural network was developed and trained thanks to its versatility in modeling inherently non-linear data. In consideration of the supervised training features of any MLP, the target was identified as the preprocessed position of the 96 anchor points. Given a tessellation configuration with a predetermined tiling position and corner point, each anchor point would be assigned to the tile to which it belonged geometrically. The MLP target would then become a set of discrete, integer pairs of coordinates.

Finally, an optimization algorithm was developed to address the main subject of this research, i.e., to investigate the possibility of reducing the positioning error through the MLP by means of an appropriate choice of the tile size and collocation of the tessellation corner point. Taking into account the number of anchor points and the possible variability of the tessellation geometry, a heuristic approach to the optimization process was selected.

A QPSO algorithm was chosen due to its features of rapid convergence to a global optimal solution and its relative ease of programming when compared to other available optimization algorithms [35].

3.3. QPSO Process

The role of the QPSO in data analysis in this research consists of searching for an appropriate set of four variables that define the tessellation properties of the search space, namely the tile width and height, and the horizontal and vertical positions with respect to the bottom-left corner point.

In most “swarm-based” optimization processes, a population of candidate solutions is first created within a multidimensional search space. Every candidate solution, also named “individual” or “particle”, is associated with a cost value, which is evaluated according to the particular optimization task. The resulting individual in this QPSO process thus becomes a four-dimensional vector of homogeneous geometrical features. At every iteration, each individual evolves its position in the search space with a law of motion linked to two attractors, namely its own best value and the overall best value that is calculated among the whole population and updated at every loop.

The QPSO algorithm, first introduced in 2004 [36,37], is inspired by the Quantum Mechanics concept that every particle is associated with a wave and the fact that quantum waves cover all the search space. The position of each individual is updated in the spirit of Quantum Mechanics and relies heavily on the generation of random numbers. The position update for each individual is governed by two attractors, the average value of all N individuals’ best positions:

$$m = \frac{1}{N} \sum_{i=1}^N b_i \quad (1)$$

and a linear combination with random coefficients c_1 and c_2 of the individual and global best:

$$p_i = (c_1 b_i + c_2 g) / (c_1 + c_2) \quad (2)$$

The position update for iteration $(n + 1)$ is then provided by the following expression, which identifies the maximum likelihood coordinated where to find the individual:

$$x_i(n + 1) = p_i + \beta(n) \cdot \text{sign}(c_3 - 0.5) \cdot \ln(c_4) |m - x_i(n)| \quad (3)$$

where c_3 and c_4 are uniformly distributed random coefficients and $\beta(n)$ is a uniformly decreasing parameter that traditionally starts at $\beta(1) = 1$ and reaches $\beta = 0.1$ after a preset number of iterations, at which the algorithm ends.

Considering that the QPSO procedure involves individuals and positions evaluated by a linear combination of vectors, its applicability is naturally taken into consideration, as in this work the search space is a subset of the multidimensional set \mathbb{R}^4 . The cost function $f(x_i)$ was evaluated for each individual $x_i = (w_i, h_i, u_i, v_i)$ using an MLP neural network, which was designed to provide the most accurate positioning achievable through appropriate network training.

3.3.1. MLP–ADAM-Based Localization

An Artificial Neural Network approach [38] is justified by the observation that the variations in each of the six meaningful signals along the research area are strongly non-linear and unpredictable due to their complexity. A commonly used ANN architecture is the simple, fully connected, feedforward MLP network, which utilizes non-linear filtering functions. MLP training is usually performed by the backpropagation technique, which is a momentum-improved gradient-descent reduction process of the regression error.

The ultimate goal of any MLP training consists of reducing the error computed between the network output and a preset target in a supervised learning procedure. The available input data is usually split into two subsets, namely the training data and the test data. The network is trained by altering the weight matrixes that characterize the MLP

architecture by using the {input vector—expected target} pairs of the training data alone. The pairs {test output—expected target} are used as a form of quality control for the training in terms of the verification of the network generalization vs. memorization capabilities.

In this work, three WiFi signals and three WSN signals were collected from 96 anchor points in a regular array of positions in the research area. All the experimental data were included in the training set, while the test set was created as a list of 20 points randomly selected within the experimental area. The test signals were generated by performing standard bilinear interpolation on the actual measured values in the vertices of the rectangle containing each of the test points. Figure 2 (left) shows a typical layout of the training and testing points within the experimental area.

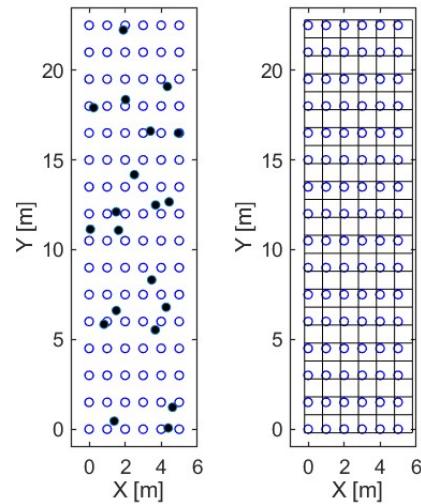


Figure 2. (left) The layout of the anchor points (blue circles) and test points in random positions (black dots) within the experimental area. (right) A tessellation of the research area is shown, with respect to the anchor points.

The preprocessing of the input data before feeding them into the MLP involved two steps. First, a relative weighting of the signal intensities was applied based on our previously published results on the same dataset [32]. Specific weights were assigned to each of the six signals. In the second step, the average of each individual signal intensity was subtracted from the weighted values to extract and retain their meaningful spatial variations using the following equation:

$$S'_j = S_j - \frac{1}{N} \sum_{i=1}^N S_{ji} \tag{4}$$

for $j = 1-6$ and $N = 116$, as the 20 test data were included in Equation (4) for consistency in the training process.

The preprocessing of the target data required the transformation of the available spatial coordinates into a discrete representation, following the concept of surface tessellation. The tile sizes and the offset position of the tessellation are the goals of the optimization process of this research. Thus, the transformation from spatial to discrete coordinates is fundamentally dependent on the individual candidate solutions, which are managed by the QPSO algorithm. Nonetheless, the preprocessing of the target followed a standard procedure in which each anchor point is univocally assigned to the tile it lies on geometrically. First, the integer number of tiles N_x and N_y are calculated, respectively, for x and y directions:

$$\begin{cases} N_x = \left\lceil \frac{x_{max}-u}{w} \right\rceil \\ N_y = \left\lceil \frac{y_{max}-v}{h} \right\rceil \end{cases} \tag{5}$$

where x_{max} and y_{max} are the maximum positional coordinates or any train or test data. The partial square bracket symbol $\lceil \cdot \rceil$ indicates the rounding up to the next integer, i.e., the number of tiles. The values u and v —offset coordinates where the tessellation starts; w, h —the width and height of the tiles are all passed to the MLP as parameters by the QPSO algorithm. Consequently, each anchor point or test point of coordinates (x_i, y_i) is associated with the discrete ‘tile’ values:

$$\begin{cases} n_x = \lceil \frac{x-x_{off}}{u} \rceil \\ n_y = \lceil \frac{y-y_{off}}{v} \rceil \end{cases} \tag{6}$$

Figure 2 (right) shows an example of virtual tessellation of the research space, in which square tiles with a side of 1 m have been superposed on the array of experimental points, with an offset of 20 cm in both directions, resulting in an array of 6×22 tiles.

The last step of the target preprocessing is required by the fact that the output of the standard MLP network lies in the interval $[0, 1]$, thus a linear normalization of the set of values n_x and n_y must be performed according to:

$$\begin{cases} n'_x = 0.1 + 0.8 \frac{n_x-1}{N_x-1} \\ n'_y = 0.1 + 0.8 \frac{n_y-1}{N_y-1} \end{cases} \tag{7}$$

This rescaling guarantees that the set of normalized targets lies in the range $[0.1, 0.9]$, thus optimizing the performance of the training process of the MLP in terms of computing speed and target separation.

The training data are used in the standard feed forward-backpropagation learning process, which aims to find appropriate weight matrices W_k that best model the neural network on the spatial distribution of the experimental data. The output of the testing data is used for the training termination criterion, whereas the testing output, calculated with the same weight matrices, is not used for training. The testing error is obtained and stored, and the MLP training process is deemed complete when the relative positioning error reaches its minimum value. The first step of the feedforward phase consists of calculating the first hidden layer h_1 using the whole input data X organized in a matrix of 96 rows—the signals in each point—and six columns—the six signals in each point—thus using the batch training process:

$$h_1 = f(XW_1) \tag{8}$$

In this expression, $f(\cdot)$ is the filter or activation function that expresses the non-linearity feature of an MLP network. In this work, we used the standard sigmoid function $f(x) = \frac{1}{1+\exp(-x)}$.

In our previous research using the same experimental data [32], we determined that the best results were achieved with three hidden layers, thus here we keep the same architecture for this work. The second and third hidden layers as well as the output are then calculated according to Equation (9):

$$\begin{aligned} h_{k+1} &= f(h_k W_{k+1}) \\ out &= f(h_3 W_4) \end{aligned} \tag{9}$$

The same process is performed separately for the test data. The MLP errors for both training and testing are calculated as $err = \frac{1}{2} \sum \sum (out - tar)^2$ where the sum extends to all elements of the output and target matrices. The standard MLP training is based on the gradient descent process in which the partial derivatives $\Delta W_k = \partial err / \partial W_k$ of the weight matrixes are calculated and the weights W_k are updated according to the expression:

$$W_k^{(n+1)} = W_k^{(n)} + \lambda \Delta W_k^{(n)} + \eta \Delta W_k^{(n-1)} \tag{10}$$

where λ is the learning rate and μ is the momentum, a term that is included to speed up the training process by taking into account the correction applied at the previous iteration.

In this work, we employed a refinement of the backpropagation algorithm, i.e., the novel adaptive momentum technique, or ‘adam’ [39], which is a proven method for expediting the learning process of the MLP. In this process, at every loop the update of each element in the weight matrices \mathbf{W}_k is proportional to the partial derivative $\partial err / \partial \mathbf{W}_{kij}$ through a variable learning rate factor, which increases the training speed in flat regions of the MLP error. Two constants, β_1 and β_2 , are introduced, as well as a ‘first’ and a ‘second’ set of momentum matrices, $\boldsymbol{\eta}$ and $\boldsymbol{\mu}$, respectively. The equations for the update of the first and second momenta are:

$$\begin{aligned}\boldsymbol{\eta}_k^{(n+1)} &= \beta_1 \boldsymbol{\eta}_k^{(n)} + (1 - \beta_1) \Delta \mathbf{W}_k \\ \boldsymbol{\mu}_k^{(n+1)} &= \beta_2 \boldsymbol{\mu}_k^{(n)} + (1 - \beta_2) (\Delta \mathbf{W}_k)^2\end{aligned}\quad (11)$$

and the weights are updated according to the expression:

$$\mathbf{W}_k^{(n+1)} = \mathbf{W}_k^{(n)} - \lambda \frac{(1 - \beta_2^n) \boldsymbol{\eta}_k^{(n+1)}}{(1 - \beta_1^n) \sqrt{\boldsymbol{\mu}_k^{(n+1)}}}\quad (12)$$

3.3.2. Positioning Algorithm

In this research work, the positioning algorithm was developed within the frame of a search space quantization, in which a regular tessellation is generated. Training and testing points are assumed to belong to a rectangular tile whose spatial coordinates are numbered with integer values.

The goal of the optimization process consists of finding optimal horizontal and vertical dimensions for the tiles, as well as an optimal starting point for the surface tessellation. For any given geometrical configuration, each experimental point belongs to a specific tile. The MLP was designed with the task of reducing—by Adam-driven descent—the error:

$$err = \frac{1}{2} \sum \sum (out - tar)^2\quad (13)$$

as per the algorithm described in Section 3.3. The process was terminated after a predetermined number of iterations when the minimum value of the real space positioning error for the test data was supposedly reached. For this reason, several steps had to be performed at regular intervals during the MLP iterations. First, the output out_{test} of the test data was calculated and rescaled by inverting the expressions in Equation (7), and the result was rounded to the nearest integer in order to identify the output tile the test data was assigned to. Then, the horizontal and vertical tile mismatches were evaluated for each test point, with the absolute difference between the output tile numbers $[n_{x,out}, n_{y,out}]$ and the target values $[n_{x,tar}, n_{y,tar}]$, which were calculated in Equation (6):

$$\begin{cases} \Delta n_x = |n_{x,out} - n_{x,tar}| \\ \Delta n_y = |n_{y,out} - n_{y,tar}| \end{cases}\quad (14)$$

Finally, the real horizontal and vertical mismatches were calculated, and the error in the test data was evaluated as the average of the distance errors in Euclidean metrics in the set of all test points:

$$err_{real} = \langle \sqrt{(w\Delta n_x)^2 + (h\Delta n_y)^2} \rangle\quad (15)$$

In order to perform a meaningful comparison between tiles of different dimensions during the optimization process, an amount was added on the basis that a point randomly located in a rectangular tile and relocated to the center carries an intrinsic error given by:

$$err_{[h,w]} = \iint \sqrt{x^2 + y^2} dx dy = \frac{1}{6} \left[\sqrt{h^2 + w^2} + \frac{h^2}{2w} \ln \frac{\sqrt{h^2 + w^2} + w}{h} + \frac{w^2}{2h} \ln \frac{\sqrt{h^2 + w^2} + h}{w} \right] \tag{16}$$

where the integral is calculated in a rectangle of dimensions $[h, w]$.

4. Analysis and Experimental Results

This section is devoted to providing detailed analysis and results obtained using the proposed algorithm.

4.1. Experimental Setup

In order to achieve the stated goal, a standard QPSO algorithm was designed with the parameters shown in Table 1, which are used as standard values capable of speeding up the overall QPSO procedure. The search range for the tile dimensions was limited to the interval (0.3, 1.2) m for both horizontal and vertical dimensions, while the offset was limited to the actual tile size to avoid placing tiles without experimental data in the lower-left corner of the research area.

Table 1. QPSO design and architecture parameters for tile size optimization.

Design Parameters		Architecture Parameters	
Number of individuals	20	c_1	1
Number of iterations	100	c_2	1.5
Number of cost calculations	2000	Initial β_0	1
Search range (tile size)	(0.3, 1.2)	Final β_0	0.1

The algorithm for the proposed technique was developed in Matlab and ran on a Windows-based 2.6 GHz Core i7 desktop. The QPSO algorithm, like any other heuristic optimization process, is supposed to provide better search results using larger populations and a greater number of generations. However, for this work, each cost calculation took an average of 50 s; thus, the values reported in Table 1 were accepted as a good compromise between computing time and the quality of the QPSO capabilities.

The outcome of any MLP is usually heavily dependent on the initial random set of the training weight matrixes. Consequently, a large amount of MLP training runs are required to identify suitable initial weight choices and achieve meaningful results.

A number of validation runs were performed to investigate the effect of the random positioning of the 20 test points inside the research area. In agreement with the investigation presented in our previous research [32], no meaningful difference was observed in the positioning error for constant test points versus random ones. Both in these validation runs and in the actual QPSO algorithm, the MLP was run 1000 times in order to achieve the best performances, i.e., to identify a suitable set of weights for which the positioning error of the test population was minimal. For each training run, the profiles of both the training and testing positioning errors versus the learning iteration were recorded, and the absolute minimum of the testing error was identified, as shown in Figure 3.

It was observed during the validation runs that the absolute minimum of the test error was most probably achieved shortly after 4000 iterations. Consequently, it was decided to stop the MLP learning run after 5000 cycles of feedforward-backpropagation as a good trade-off between accuracy and total computing time. The cost value for each QPSO individual was evaluated as the average of the best 50 of 1000 MLP runs, which would correspond to fortunate initial random choices of the weights.

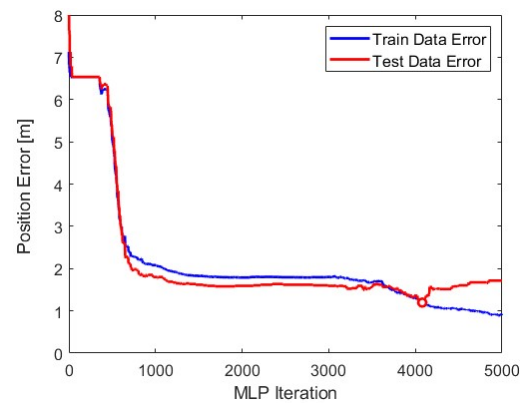


Figure 3. The profiles of the position error relative to the training and testing data in a typical MLP run. The minimum value in the test profile is highlighted.

4.2. Results

Following the definition of the optimization procedure using a QPSO algorithm and the technique for calculating the localization error using a standard MLP with an adaptive backpropagation process, the next step in this research was to select the most appropriate values for both the QPSO and MLP algorithms. Several parameters can affect the speed of both processes and strongly alter the total computing time. For the QPSO part, it was decided to choose the standard parameters found in the literature, namely those reported in Table 1. The QPSO procedure was repeated for a total of 100 generations, each producing a new population of 20 vectors of 4 components, recording the size of the tiles as well as the offset position with respect to the bottom left experimental point.

Due to the QPSO's nature, the global best individual, i.e., the vector of signal weights producing the lowest positioning error, was recorded at every stage of the algorithm. Each vector was first tested under the preset conditions, which limited the tile size to the interval (0.3, 1.2) m for both horizontal and vertical dimensions and the offset to the tile size. Each non-compliant configuration was discarded and the generation of a new individual was repeated until the preset population size was reached. The positioning errors of the new QPSO generation of individuals were then evaluated by repeating 1000 times the MLP procedure for each individual and calculating the average of the 50 best test positioning errors.

Multilayer Perceptron neural networks are subject to several parameters that greatly affect the training accuracy and the computing time. For this research, it was decided to refer to the investigation of the validation runs in our previously published research on the same dataset [32] and to keep the same MLP architecture. In particular, three hidden layers of 18 hidden units each were retained, as well as the standard sigmoid activation function. The values of the learning rates and momentum coefficients for the Adam technique were determined by visual inspection during a small number of dry runs, as well as by measuring the computing time of individual cost calculations. Moreover, the intensities of the input signals were weighted, in agreement with our previous results [32]. The set of parameters for the MLP process are reported in Table 2. Because of the sheer amount of QPSO cost evaluations and the large number of MLP runs for each individual, the overall computing time reached about 28 h despite the fact that each MLP training process of 5000 iterations took only 50 ms to complete.

The output of the MLP process was designed to return the integer values of the tile positions in the regular array defined by the vectors $x_i = (w_i, h_i, u_i, v_i)$, both for the train and the test data points. Equations (14)–(16) were then applied to calculate the actual position error for both data, while only the test position error was retained as the result of this research.

Table 2. MLP design and operational parameters.

Architecture Parameters		Training Parameters	
Hidden layers	3	Learning rate	$\eta = 0.001$
Hidden units	18 per layer	Momenta Adam values	$\beta_1 = 0.9, \beta_2 = 0.999$
Activation function	Logistic sigmoid	Runs per individual	1000
Training points	96	Training iterations	5000
Test points	20	Computing time	~28 h

A meaningful comparison of configurations with different tile sizes cannot be done on the basis of the number of tiles mismatch in either direction alone, as larger tiles might be correctly identified despite a relevant actual position error. For this reason, the value of the intrinsic error was taken into account as the average distance between a generic point and the center of the tile, according to Equation (16). Thus, the actual positioning error err_{pos} , i.e., the QPSO cost value for an individual becomes:

$$err_{pos} = err_{real} + err_{[h,w]} \tag{17}$$

in which the two parts are defined by Equations (15) and (16).

The value of the resulting positioning error as a function of the QPSO iteration number is shown in Figure 4 (right). It can be observed that the improvements in the positioning error take place in stages, in agreement with the discrete exploration technique of the QPSO. No error improvement was achieved beyond QPSO iteration #56, which provided a positioning error m measured on test data alone. This value shows an improvement of ~16% with respect to our previous best result, in which signal weighting was first introduced for the analysis of the same dataset.

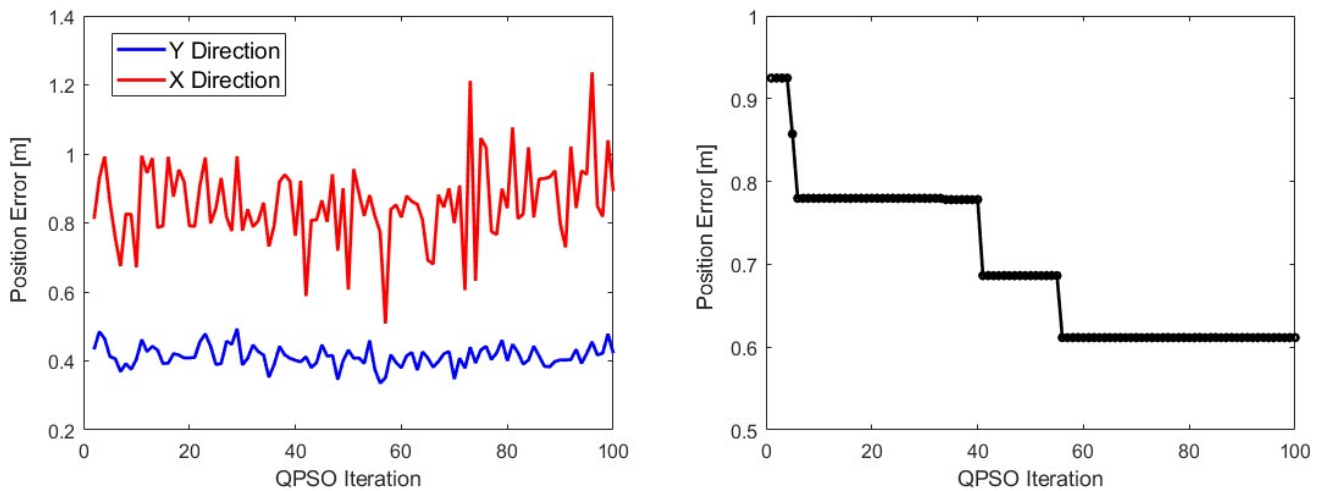


Figure 4. (left) The values of the horizontal and vertical contributions to the total positioning error, as calculated at each QPSO iteration. (right) The overall performance of the QPSO algorithm in terms of the positioning error reduction. A final value of 0.611 m was achieved.

The final geometrical configuration that led to this positioning error provided a tile size $(w, h) = (0.505, 0.587)$ m and an offset $(u, v) = (0.059, 0.411)$ m. These values correspond to a tessellation of 11 tiles horizontally and 40 vertically. The tile size provides an intrinsic error of 0.209 m as per Equation (16), thus leading to a tile error of 0.402 m. In consideration of the tile size and the procedure of calculating the positioning error, it can be concluded that, on average, each test data carries a mismatch of a single tile.

In our previously published work on the same dataset using fixed surface tessellation [27], it was observed that the positioning error was mainly caused by tile mismatches in the horizontal direction. For this reason, in this research, further attention was given to the separate contributions to the total positioning error by the horizontal and vertical tile

mismatches. At every QPSO iteration, the position error in each direction was recorded for the best individual out of the population of 20. Again, it can be observed that in the horizontal direction, the partial position error is larger and more widely distributed compared to the values recorded in the vertical direction. Figure 4 (left) shows that in the vertical direction, the error does not vary much around 40 cm, while in the horizontal direction, values are spread in the range of 0.6–1.2 m. Nevertheless, the particularly low value recorded at QPSO iteration #56 generated the final global best for this research.

A comparison of the results obtained in this research with a selection of relevant previously published location average errors is presented in Table 3. Our previous work, based on the same raw data, is included together with other benchmark results available in the literature, showing a remarkable reduction in the positioning incertitude.

Table 3. Comparison of our obtained results with previously published research.

Reference	Chosen Approach and Implementation	Research Area and Anchor Points (AP)	Error
M. Ficco et al. [40]	Source selection, WiFi + Bluetooth	20 m × 21 m, 26 AP	>4 m
L. Jiang [26]	K-NN modeling, WiFi + RSS	15 m × 40 m, 100 AP	1.70 m
Z. Xiong et al. [41]	Particle filter, WiFi + RFID	25 m × 12 m, 80 AP	1.60 m
J. Chen et al. [28]	Optimization, IMU + WLAN	90 m path, 113 AP	1.48 m
Z. Farid et al. (same dataset) [33]	MLP modeling, WiFi + WSN	6 m × 24 m, 96 AP	1.22 m
I. U. Khan et al. (same dataset) [27]	Tessellation, WiFi + WSN	6 m × 24 m, 96 AP	1.01 m
E. Scavino et al. (same dataset) [32]	Optimization, WiFi + WSN	6 m × 24 m, 96 AP	0.725 m
This Research (same dataset)	Variable tessellation, WiFi + WSN	6 m × 24 m, 96 AP	0.611 m

The proposed method of data analysis does not present any particular challenges, apart from the extensive computing time. Nevertheless, the actual application of this technology would require some caveats about the stability and availability of the sources of the signals. Possible drifts and abrupt variations would render this method less accurate, thus requiring particular attention to the hardware of the whole installation. Further work beyond this method should possibly focus on the normalization of the input signals and the analysis of their spatial distributions.

5. Conclusions

This paper illustrates a QPSO algorithm for an indoor location system based on hybrid raw RSS signals, namely those from WiFi and WSN sources. The proposed system considered the possibility of reducing the positioning error by performing a regular tessellation of the research area in which the raw signals were acquired. Each QPSO individual was identified as a vector of four values containing the tile size and offset, which unambiguously determined a tessellation geometry. A Multilayer Perceptron network was developed as a tool to heuristically predict the positioning error for each QPSO individual. The architecture of the MLP, as well as the initial raw signal weighting, were kept in agreement with our previous research, thus allowing a faster network setup and positioning error reduction. The final results of the optimization process allowed us to identify an optimal geometrical configuration of the regular surface tessellation in which the positioning error showed a marked reduction compared to our previous research using the same set of data. We achieved a final localization accuracy of 0.611 m, representing an improvement of at least 16% compared to our previous best results using the same dataset. This value also marks an improvement of at least 50% over our initial published analysis of these data.

In addition to determining the overall positioning accuracy, a raw analysis of the spatial source of error was performed in order to identify the contribution along the horizontal and vertical directions. For every QPSO generation, in the horizontal direction, the error component was larger by roughly a factor of two compared to the vertical component, thus providing a large amount of the positioning error.

The goal of any indoor positioning algorithm consists of reducing the localization error in real-life scenarios in which randomly set wireless receivers are capable of precisely inferring their spatial coordinates. To this date, we have analyzed hybrid approaches with two sources, namely WiFi and WSN, and several algorithms, i.e., signal weighing with smooth research space and variable tessellation in this work. These results encourage further investigation, and in future work, all the developed approaches may be joined together, as well as including further techniques of spatial data analysis or an improved MLP training process that takes into account the different directional error contributions.

Author Contributions: Conceptualization, E.S., M.A.A.R., Z.F., S.A., and M.A.; Data curation, M.A.A.R., Z.F., and M.A.; Formal analysis, E.S. and Z.F.; Funding acquisition, S.A. and M.A.; Investigation, M.A.A.R., S.A., and M.A.; Methodology, E.S. and M.A.A.R.; Resources, M.A.A.R., Z.F., and M.A.; Software, E.S.; Supervision, M.A.A.R.; Validation, M.A.A.R., Z.F., S.A., and M.A.; Visualization, E.S. and M.A.; Writing—original draft, E.S.; Writing—review and editing, M.A.A.R., Z.F., S.A., and M.A. All authors have read and agreed to the published version of the manuscript.

Funding: This research was supported by EIAS Data Science & Blockchain Lab, Prince Sultan University. The authors would also like to thank Prince Sultan University for paying the APC for this article.

Data Availability Statement: Data can be shared upon reasonable request from the corresponding author.

Acknowledgments: The authors would like to thank Prince Sultan University for their support. In addition, the authors also wish to thank Rosdiadee Nordin and Mahamod Ismail of UKM for helping with the organization of data collection in public areas of the Faculty of Engineering.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Wang, H.; Ning, H.; Lin, Y.; Wang, W.; Dhelim, S.; Farha, F.; Ding, J.; Daneshmand, M. A Survey on the Metaverse: The State-of-the-Art, Technologies, Applications, and Challenges. *IEEE Internet Things J.* **2023**, *10*, 14671–14688. [[CrossRef](#)]
2. Qays, M.O.; Ahmad, I.; Abu-Siada, A.; Hossain, M.L.; Yasmin, F. Key communication technologies, applications, protocols and future guides for IoT-assisted smart grid systems: A review. *Energy Rep.* **2023**, *9*, 2440–2452. [[CrossRef](#)]
3. Mehannaoui, R.; Mouss, K.N.; Aksa, K. IoT-based food traceability system: Architecture, technologies, applications, and future trends. *Food Control* **2023**, *145*, 109409. [[CrossRef](#)]
4. Saleem, F.; Wyne, S. WLAN-based indoor localization using neural networks. *J. Electr. Eng.* **2016**, *67*, 299–306. [[CrossRef](#)]
5. Rahman, M.A.A.; Karim, M.K.; Bundak, C.E. Weighted local access point based on fine matching k-nearest neighbor algorithm for indoor positioning system. In Proceedings of the 2019 AEIT International Annual Conference, Florence, Italy, 18–20 September 2019; pp. 1–5.
6. Gupta, D.; Wadhwa, S.; Rani, S.; Khan, Z.; Boulila, W. EEDC: An Energy Efficient Data Communication Scheme Based on New Routing Approach in Wireless Sensor Networks for Future IoT Applications. *Sensors* **2023**, *23*, 8839. [[CrossRef](#)] [[PubMed](#)]
7. Khan, A.; Abdeljawad, T.; Alqudah, M.A. Neural networking study of worms in a wireless sensor model in the sense of fractal fractional. *AIMS Math.* **2023**, *8*, 26406–26424. [[CrossRef](#)]
8. Ebied, M.; Elmisery, F.A.; El-Hag, N.A.; Sedik, A.; El-Shafai, W.; El-Banby, G.M.; Soltan, E.; Al-Zubi, N.; Brisha, A.; Zekry, A.; et al. A Proposed Deep-Learning-Based Framework for Medical Image Communication, Storage and Diagnosis. *Wirel. Pers. Commun.* **2023**, *131*, 2331–2369. [[CrossRef](#)]
9. Realini, E.; Caldera, S.; Pertusini, L. Precise GNSS positioning using smart devices. *Sensors* **2017**, *17*, 2434. [[CrossRef](#)] [[PubMed](#)]
10. Yeh, S.; Hsu, W.; Lin, W.; Wu, Y. Study on an indoor positioning system using Earth's magnetic field. *IEEE Trans. Instrum. Meas.* **2020**, *69*, 865–872. [[CrossRef](#)]
11. Tao, Y.; Wu, L.; Sidén, J.; Wang, G. Monte Carlo-based indoor RFID positioning with dual-antenna joint rectification. *Electronics* **2021**, *10*, 1548. [[CrossRef](#)]
12. Scherhäufl, M.; Rudic, B.; Stelzer, A.; Pichler-Scheder, M. A blind calibration method for phase-of-arrival-based localization of passive UHF RFID transponders. *IEEE Trans. Instrum. Meas.* **2019**, *68*, 261–268. [[CrossRef](#)]
13. Zhou, J.; Ke, Y.; Yu, K.; Tian, D. A solution of high-precision WLAN positioning based on TDOA and PTP. *MATEC Web Conf.* **2016**, *61*, 7018. [[CrossRef](#)]
14. Alshami, I.H.; Ahmad, N.A.; Sahibuddin, S.; Firdaus, F. Adaptive indoor positioning model based on WLAN-fingerprinting for dynamic and multi-floor environments. *Sensors* **2017**, *17*, 1789. [[CrossRef](#)] [[PubMed](#)]
15. Fang, K.T.; Lee, C.T.; Sun, L. An improved hierarchical WLAN positioning method based on apriori knowledge. *Electronics* **2019**, *8*, 475. [[CrossRef](#)]

16. Bencak, P.; Hercog, D.; Lerher, T. Indoor positioning system based on Bluetooth low energy technology and a nature-inspired optimization algorithm. *Electronics* **2022**, *11*, 308. [[CrossRef](#)]
17. Chen, J.; Wang, S.; Ouyang, M.; Xuan, Y.; Li, K.C. Iterative positioning algorithm for indoor node based on distance correction in WSNs. *Sensors* **2019**, *19*, 4871. [[CrossRef](#)] [[PubMed](#)]
18. Benatia, M.A.; Sahnoun, M.; Baudry, D. Multi-objective WSN deployment using genetic algorithms under cost, coverage, and connectivity constraints. *Wirel. Pers. Commun.* **2017**, *94*, 2739–2768. [[CrossRef](#)]
19. Li, T.; Yan, W.; Ping, L.; Fang, P. A WSN positioning algorithm based on 3D discrete chaotic mapping. *EURASIP J. Wirel. Commun. Netw.* **2019**, *126*, 1323. [[CrossRef](#)]
20. Yanbin, H.; Xiaodong, Y.; Abbasi, Q.H. Efficient AoA-based wireless indoor localization for hospital outpatients using mobile devices. *Sensors* **2018**, *18*, 3698. [[CrossRef](#)]
21. De Gante, A.; Siller, M. A survey of hybrid schemes for location estimation in wireless sensor networks. *Procedia Technol.* **2013**, *7*, 377–383. [[CrossRef](#)]
22. Abed, A.; Abdel-Qader, I. RSS-fingerprint dimensionality reduction for multiple service set identifier-based indoor positioning systems. *Appl. Sci.* **2019**, *9*, 3137. [[CrossRef](#)]
23. Arai, H. Antennas in Access Points of WLAN/WiFi. In *Handbook of Antenna Technologies*; Chen, Z., Liu, D., Nakano, H., Qing, X., Zwick, T., Eds.; Springer: Singapore, 2016; pp. 2579–2588.
24. Ali, R.A.E.Y.; Babiker, A.; Mustafa, A.N. Performance analysis of WiFi network. *IOSR J. Electron. Commun. Eng.* **2015**, *10*, 67–76.
25. Farid, Z. Higher Accuracy Hybrid Wireless Indoor Localization Using Machine Learning. Ph.D. Thesis, Faculty of Engineering, Universiti Kebangsaan Malaysia, Bangi, Malaysia, 2016.
26. Jiang, L. A WLAN Fingerprinting Based Indoor Localization Technique. Master's Thesis, Department Computer Science, University of Nebraska, Lincoln, NE, USA, 2012.
27. Khan, I.U.; Ali, T.; Farid, Z.; Scavino, E.; Rahman, M.A.A. An improved hybrid indoor positioning system based on surface tessellation artificial neural network. *Meas. Control* **2020**, *53*, 1968–1977. [[CrossRef](#)]
28. Chen, J.; Song, S.; Yu, H. An indoor multi-source fusion positioning approach based on PDR/MM/WiFi. *AEU—Int. J. Electron. Commun.* **2021**, *135*, 153733. [[CrossRef](#)]
29. Luo, J.; Zhang, Z.; Liu, C.; Luo, H. Reliable and cooperative target tracking based on WSN and WiFi in indoor wireless networks. *IEEE Access* **2018**, *6*, 24846–24855. [[CrossRef](#)]
30. Guo, T.; Chai, M.; Xiao, J.; Li, C. A hybrid indoor positioning algorithm for cellular and Wi-Fi networks. *Arab. J. Sci. Eng.* **2022**, *47*, 2909–2923. [[CrossRef](#)]
31. Gang, H.-S.; Pyun, J.-Y. A smartphone indoor positioning system using hybrid localization technology. *Energies* **2019**, *12*, 3702. [[CrossRef](#)]
32. Scavino, E.; Rahman, M.A.A.; Farid, Z. An improved hybrid indoor positioning algorithm via QPSO and MLP signal weighting. *Comput. Mater. Contin.* **2023**, *74*, 379–397. [[CrossRef](#)]
33. Farid, Z.; Khan, I.U.; Scavino, E.; Rahman, M.A.A. A WLAN fingerprinting based indoor localization technique via artificial neural network. *IJCSNS Int. J. Comput. Sci. Netw. Secur.* **2019**, *19*, 157–165.
34. Farid, Z.; Nordin, R.; Ismail, M.; Abdullah, N.F. Hybrid indoor-based WLAN-WSN localization scheme for improving accuracy based on artificial neural network. *Mob. Inf. Syst.* **2016**, *2016*, 6923931. [[CrossRef](#)]
35. Xiao, P.Y.; Peng, J.; Zhou, G. An improved QPSO algorithm based on social learning and Lévy flights. *Syst. Sci. Control. Eng.* **2018**, *6*, 364–373.
36. Sun, J.; Feng, B.; Xu, W. Particle swarm optimization with particles having quantum behaviour. In Proceedings of the 2004 Congress on Evolutionary Computation, Portland, OR, USA, 19–23 June 2004; pp. 325–331.
37. Bhatia, A.S.; Saggi, M.K.; Zheng, S. QPSO-CD: Quantum-behaved particle swarm optimization algorithm with Cauchy distribution. *Quantum Inf. Process.* **2020**, *19*, 345. [[CrossRef](#)]
38. Russell, S.; Norvig, P. *Artificial Intelligence—A Modern Approach*, 4th ed.; Pearson Prentice Hall: Hoboken, NJ, USA, 2020.
39. Kingma, D.P.; Ba, J.L. Adam: A method for stochastic optimization. In Proceedings of the International Conference on Learning Representations (ICLR), San Diego, CA, USA, 7–9 May 2015; pp. 1–13.
40. Ficco, M.; Palmieri, F.; Castiglione, A. Hybrid indoor and outdoor location services for new generation mobile terminals. *Pers. Ubiquitous Comput.* **2014**, *18*, 271–285. [[CrossRef](#)]
41. Xiong, Z.; Song, Z.; Scalera, A.; Ferrera, E.; Sottile, F.; Brizzi, P.; Tomasi, R.; Spirito, M.A. Hybrid WSN and RFID indoor positioning and tracking system. *EURASIP J. Embed. Syst.* **2013**, *1*, 6. [[CrossRef](#)]

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.