



HAL
open science

UWB Time of Flight based indoor IoT localization solution with Deep Learning optimized by meta-heuristics

Sihem Tlili, Sami Mnasri, Thierry Val

► **To cite this version:**

Sihem Tlili, Sami Mnasri, Thierry Val. UWB Time of Flight based indoor IoT localization solution with Deep Learning optimized by meta-heuristics. International Journal of Sensor Networks, In press. hal-04265613

HAL Id: hal-04265613

<https://ut3-toulouseinp.hal.science/hal-04265613v1>

Submitted on 31 Oct 2023

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

UWB Time of Flight based indoor IoT localization solution with Deep Learning optimized by meta-heuristics

Abstract: Nowadays, indoor localization is among the most important challenges in IoT networks. On the other hand, Deep Learning techniques are emerging as a leading method. Additionally, meta-heuristic algorithms attract several research domains due to its efficiency in resolving optimization problems. In this work, a Deep Learning model optimized by meta-heuristic algorithms and based on Time of Flight (ToF) measurements captured by Ultra-Wide Band technology, as an indoor IoT localization solution, is proposed. The findings showed that optimization with the Grey Wolf Optimizer can accelerate convergence towards optimal parameters during the learning phase. Furthermore, it was observed that ToF measurements enhance the positioning estimation capabilities, in comparison with RSSI and Range measurements. Compared with an existing multi-lateration method, the suggested solution provided more accurate positions of the IoT mobile object as it yielded better results in terms of localization accuracy (98.92%), Mean Absolute Error (0.057m) and Mean Squared Error (0.0095m).

Keywords: Internet of Things; Indoor Localization; Meta-heuristic Optimization; Ultra-Wide Band; Time of Flight; Grey Wolf Optimizer; Arithmetic Optimization Algorithm.

1 Introduction

An indoor IoT localization solution involves applying an indoor positioning technique capable of providing precise object positioning in an IoT environment. Incorporating localization data can optimize IoT services and applications. Therefore, the development of localization solutions and methods is necessary for the proper functioning of these applications. However, indoor localization is among the most significant challenges faced by IoT networks. Indeed, users of IoT applications seek indoor positioning solutions that can give precise positioning of people and objects. However, the nature of indoor environments (walls and obstacles), prevent the radio signals emitted by satellites from penetrating these environments, making the use of satellite positioning systems difficult. In fact, for outdoor localization, Global Navigation Satellite Systems (GNSS) can be employed to provide real-time positioning of an element. These worldwide coverage systems include GPS (Global Positioning System), Galileo (Europe's Global Navigation Satellite System), Glonass (Russia's Global Navigation Satellite System) and BDS (China's BeiDou Navigation Satellite System). The limited coverage of these systems in indoor environments necessitates the development of alternative solutions.

Therefore, in the literature, several indoor positioning systems were introduced. They are often classified as range-based and range-free methods. In range-free techniques, localization is established between objects by relying on the neighborhood information. On the other hand, range-based techniques are based on measurements taken in the studied environment, such as Time of Flight (ToF), Angle of Arrival (AoA) or

Received Signal Strength Indicator (RSSI). For example, Ultra-Wide Band (UWB) ToF measurements provide high accuracy in the localization systems. Indeed, UWB [1] is considered as the most promising technology used in critical applications that require high precision.

Multi-lateration methods, which are the simplest and most popular techniques, employ the geometric methods to calculate the estimated positions. However, these algorithms have certain limitations. In practice, they provide imperfect measurements, leading to poor accuracy in the positions of mobile objects that can be estimated only using large number of fixed anchors. Therefore, it is interesting to explore the potential of artificial intelligence, specifically Deep Learning (DL), in order to estimate mobile object localization. Furthermore, meta-heuristic optimization is among the successful paradigms applied in many fields such as economics, electrical, civil and mechanical engineering.

Therefore, in the present work, a solution for indoor localization of mobile objects is proposed. It leverages three advantageous elements: Time of Flight (ToF) measurement technique based on Ultra-Wide Band (UWB) communication technology, Deep Learning and meta-heuristic optimization. Indeed, this work focuses primarily on a set of contributions related to this targeted solution. Its main objectives are presented below:

- Building DL models for indoor localization using a dataset composed, essentially, of UWB ToF measurements.
- Modeling the training of a DL model as an optimization problem by establishing a correspondence between the training of neural

layer parameters and an optimization problem resolution through meta-heuristic algorithm.

- Developing optimizers that can be used for DL model training. Each optimizer is based on a recent and widely-used meta-heuristic optimization algorithm.
- Developing and training the constructed models and testing them using the proposed optimizers.
- Evaluating the performance of the developed DL models and select the most performant among them as the suggested solution.
- Conducting a comparative study by:
 - Comparing the results obtained by DL models trained with those given by the different developed optimizers and the Gradient Descent (GD) optimizer.
 - Comparing the findings provided using ToF measurements with those obtained employing other measurements: RSSI and Range.
 - Comparing the results given by the proposed solution with those obtained by applying a multi-iteration-based localization solution.

The remaining of this article is divided into the following sections: Section 2 presents a review of the recent works that suggested solutions for indoor IoT localization. Section 3 provides the background information of this study. Section 4 explains the different contributions of the introduced proposal and the choices made to implement the developed models. Section 5 presents the results of the various experimental tests conducted to evaluate the proposal and provides details about the conducted comparative study. Section 6 summarizes the study and outlines some future research directions.

2 Related Works

In the literature, several approaches were proposed to enhance the accuracy of indoor localization in IoT networks.

For instance, [2] suggested a solution that uses LoRaWAN technology and neural networks to predict positions within a university building. The RSSI and signal-to-noise ratio (SNR) metrics were employed to measure the signal strength. A position was estimated based on the variation of the signal power. Real-world experiments were carried out to test and evaluate the developed system. The obtained results demonstrate that the latter achieved an accuracy of up to 98.8%. However, the precision rate and the accuracy of localization given by the proposed approach were not compared with those provided by other solutions that use the RSSI and SNR metrics.

Besides, [3] employed Wi-Fi RSSI measurements to train an ensemble model composed of a DNN model and two other models built using machine learning techniques (K-nearest neighbors (KNN) and Random Forest (RF)). The best model, which yielded the smallest prediction error for a test data point, was selected by an intermediate classifier. The results provided by the proposed ensemble model are better than those given by each individual base model if used separately. Indeed, the average error obtained by the introduced model during testing was 1.10m. However, in [4], a sub-meter level accuracy was required in indoor positioning systems. Despite the importance of the developed model, it was not sufficiently effective to overcome the challenge of indoor localization.

Another deep learning-based approach for indoor localization was suggested in [5]. The proposed solution used RSSI signals from BLE sensors to construct a 2D image employed to develop a CNN model. Moreover, an improved version of the PSO optimization algorithm was applied to enhance the training of the neural network. Specifically, the weights of the CNN layers were determined as the best global solutions obtained by PSO. Real-world experiments were conducted in a building to evaluate the prediction results provided by the developed framework. These findings were compared with those obtained using other machine learning techniques such as ANN, DNN, KNN, CNN, etc. The comparison demonstrated that the introduced solution offered improved localization accuracy for a mobile object, compared to other techniques, with a localization accuracy reaching 97.92%. However, it is important to note that this accuracy is based on the results provided during the validation stage. Despite their importance, these results should be confirmed by a testing phase.

Poucet [6] is an indoor localization solution designed for firefighters and soldiers. It combines Ultra-Wideband (UWB) and Time of Flight (ToF) technologies with GPS and LoRa. Trilateration was applied in Poucet development to estimate the positions of objects. The simulation results showed an average error of 0.36m.

In [7], the authors used the measurements of Time of Flight (ToF) obtained from UWB signals to develop a localization solution. The approach utilized an algorithm based on ToF measurements and combined it with the Maximum Likelihood Estimation (MLE) method to resolve the localization problem. Moreover, real-world experiments were conducted to test the performance of the presented system. The average obtained error was 0.2m.

In [8], a framework was presented as an indoor localization solution. It was formed by collecting data about CSI signals using WiFi technology. These data were employed to train a CNN (Convolutional Neural Network) model. The latter achieved an accuracy of 68.5% and an average distance error of 1.75m. However, this accuracy remains unsatisfactory for indoor IoT localization requirements. Additionally, the suggested

solution did not improve the training algorithm of the DL model.

In [9], the authors utilized an emulator software to generate a dataset of RSSI measurements. This dataset was, then, employed to train and test a MLP (Multi-Layer Perceptron) deep learning model. The results of the conducted tests revealed that the accuracy during the training and the testing phase attained 83%. However, the results should be experimentally validated on data collected in a real-world environment.

Deep Learning-Based Cooperative Architecture (DELTA) was presented in [10]. The dataset used to train the deep learning model were constructed using simulation software based on the measured RSSI signals. The development and training of the DNN (Deep Neural Network) were performed utilizing Keras. The performance of the proposed model was compared with that of two machine learning techniques: Support Vector Machine (SVM) and K-Nearest Neighbor (KNN). The obtained results showed that DELTA provided the best results with an average error of 1.6m. However, the efficiency of the introduced model should be validated in real-world environment and in an additional testing phase. Moreover, the achieved accuracy did not satisfy the requirements of IoT networks for indoor positioning as the average distance error should be inferior to 1m [4].

In [11], the UWB (Ultra-Wideband) technology was used to develop and present a distributed and collaborative localization solution. Simulations were conducted to evaluate the proposed algorithm. The obtained results revealed that the localization accuracy improved as the network density increased, and the average position error was below 1m with the average connectivity. However, real-world experiments are needed to validate these results.

In [12], RSSI and CSI measurements were collected to construct a dataset utilized to develop and test a deep learning model. The applied approach consists in developing neural networks implemented with Multi-Layer Perceptron (MLP) and Convolutional Neural Network (CNN) architectures to predict the position of a person in an indoor environment. The prediction was formulated as a classification problem for localization. The tests were conducted to compare four localization methods: MLP-RSS, MLP-CSI, CNN-RSS and CNN-CSI. The Keras platform [13] was used to apply and test these methods. The obtained results demonstrated that the CNN-CSI method outperformed the other localization techniques by achieving a distance error of 0.92m.

Another indoor positioning solution in an IoT environment was presented in [14]. It involves training a CNN model using the measured RSSI signals. Specifically, the RSSI signals were employed to construct 3D radio images, and the localization problem was formulated as an image recognition problem using these constructed images as input data of the developed deep learning model. Simulations were conducted to test

the proposed model and compare its performance with those of other approaches that utilize RSSI signals to predict the positions. The experimental findings showed that, with an average accuracy of up to 94.13%, the performance of the introduced model was the best, compared to those of the other approaches. However, it is important to note that this accuracy was obtained when applying the developed approach in a simulated IoT environment. In other words, it is necessary to evaluate the performance of the proposed system in a real-world environment.

In [15], the trilateration technique was applied to propose a localization method. The proposal used RSSI measurements to estimate, first, the distances between objects and anchors and, then, predict the positions of the objects. Simulations were conducted to evaluate the performance of the introduced system that provided an average localization error of 1.37m.

The trilateration technique was also used in [16] to develop and present a localization algorithm. The solution is based on Time of Flight (ToF) measurements captured by UWB technology. It involves two steps. In the first stage, ToF measurements were used to estimate the distances between the object to be localized and a set of anchors. However, the second step consists in employing the actual positions of the anchors and the estimated distances to calculate the position of the object applying a specific trilateration algorithm. The conducted experiments demonstrated an average error of around a dozen centimeters.

Moreover, a localization framework relying on Deep Neural Network (DNN) and CSI signals was introduced to predict the positions of IoT objects in [17]. The designed framework was tested in two different indoor environments. During the tests, the results obtained with CSI measurements were compared to those provided by RSSI measurements. It was also obvious that the accuracy achieved using CSI was higher than that given by RSSI. Additionally, it was demonstrated that the proposed framework outperformed the SVM technique in terms of localization accuracy. The average obtained error ranged from 0.6 to 0.7 meters for the two tested environments.

PSO-ANN combines the Artificial Neural Network (ANN) technique and the Particle Swarm Optimization (PSO) algorithm. This technique was applied by [18] to develop an indoor positioning system. In addition to PSO and ANN, the developed approach used the "Affinity Propagation (AP)" clustering algorithm to obtain better position prediction. Initially, RSSI data was collected from known reference points. Then, the clustering algorithm was employed to divide the reference points in the studied environment into clusters. Subsequently, an ANN model was trained for each cluster. It was noticed that developing a suitable ANN model for each cluster helped to estimate more accurate positions. The proposed system was implemented on a mobile phone to test the hybrid strategy on a building floor. The position prediction results were

compared with those obtained by applying ANN and KNN with and without clustering. The comparison demonstrated that the hybrid ANN+PSO+AP strategy outperformed the other studied techniques in terms of localization accuracy. Furthermore, it showed that clustering improved the accuracy of the predicted positions.

Table 1 summarizes the works described in this section based on the following criteria: (1) The year in which the studied method was proposed; (2) The technique applied to resolve the localization problem; (3) Whether the employed technique was optimized (4) Whether a meta-heuristic optimization algorithm was applied (5); Communication technology and (6) Measurement techniques (the communication technology and the nature of the signals employed during data collection); (7) The average distance error (a performance indicator that calculates the average distance between the actual position and the estimated one); (8) The accuracy of the obtained localization results; (9) The time required to estimate a position, if mentioned; and (10) Experiments (the nature of the conducted experiments on which the collected data and the results are based (real experiments or simulations)).

The survey of the above-mentioned works revealed that the existing methods cannot meet the necessary localization accuracy in IoT environments. Indeed, the average distance error they provided between the actual position and the estimated one is a few tens of centimeters. Furthermore, no technique that uses UWB ToF data was introduced to train a Deep Learning model and predict the position of objects in IoT environments, despite the fact that UWB is the most promising technology for indoor localization systems and ToF values can be accurately measured based on UWB signals.

On the other hand, it was observed that the use of the PSO optimization algorithm with Deep Learning models improved the performance and accuracy of the localization method. Taking advantage of the success of the meta-heuristic optimization algorithms and their high efficiency in solving complex problems, it is necessary to model a more efficient solution for indoor IoT localization based on other optimization algorithms that are more recent and performant than PSO. Additionally, no proposal that focuses on solving indoor localization in IoT networks by training DL models with the Grey Wolf Optimizer (GWO) and Arithmetic Optimization Algorithm (AOA) was introduced in the literature. However, GWO and AOA remain the most recent and promising meta-heuristic optimization algorithms.

In the present study, UWB Time of Flight measurements are employed to achieve better accuracy for the localization of mobile objects in an IoT environment by combining the advantages of meta-heuristic optimization algorithms and those of Deep Learning methods. The following section presents the context of the current work.

3 Background

This section concentrates on the necessary concepts utilized to establish the context of this study. It introduces the most commonly-used techniques for indoor IoT localization. It, then, lists the most popular Deep Learning algorithms and the meta-heuristic optimization algorithms.

3.1 Indoor IoT localization

Generally, the localization techniques can be classified into two types: range-based and range-free methods.

3.1.1 Range-free localization techniques

Range-free localization techniques often provide an estimated position without using anchors. They typically involve collaborative methods as the estimation of positions relies on the neighborhood information and the geometric methods. Their main advantage is that they do not require special hardware support, making them less costly. The DV-Hop [19] and Centroid [20] algorithms are among the most popular range-free techniques.

3.1.2 Range-based localization techniques

Range-based localization techniques are based on measurements taken in the studied IoT environment to obtain distance or angle estimations. These techniques offer high precision, but they require expensive hardware and more challenging deployment. The most commonly-applied measurement techniques in the literature [21] are:

- Received Signal Strength Indicator (RSSI) [22]: measures the power of the signal received during a transmission and, therefore, estimates the distance between the transmitter and the receiver. RSSI measurements do not require additional hardware other than a device using wireless communication technology. Therefore, RSSI is the most widely-used measurement technique applied to implement the localization strategies in indoor environments. However, it is less precise in terms of localization accuracy because the received power is often disturbed by obstacles and non-omnidirectional antennas.
- Channel State Information (CSI): combines information about the communication channel's state and describes how a message propagates from the transmitter to the receiver. In [23], the authors proved that CSI provides better detection accuracy than RSSI. However, this technique is less performant than the Time of Flight (ToF) radio technique.
- Angle of arrival (AoA): measures the angle at which the signal arrives at the receiver. For

Table 1 Summary of the studied works

Paper	Year	Technique	Optimization of employed technique	Meta-heuristic optimization	Technology	Measurement techniques	Results		Estimation time (milliseconds)	Experiments
							Average distance error (meter)	Accuracy		
[2]	2023	DNN	✗	✗	LoRaWAN	RSSI et SNR	-	98.8%	-	real
[3]	2021	Ensemble learning (KNN, DNN and RF)	✗	✗	WiFi	RSSI	1.10	-	-	real
[5]	2021	CNN	✓	enhanced version of PSO	BLE	RSSI	-	97.92%	-	real
[6]	2021	Trilateration	✗	✗	UWB, LoRa	ToF	0.36	-	-	simulations
[7]	2021	MLE method	✗	✗	UWB	ToF	0.2	-	-	real
[8]	2020	CNN	✗	✗	WiFi	CSI	1.75	68.5%	1.89	real
[9]	2020	MLP	✗	✗	WiFi	RSSI	-	83%	-	simulations
[10]	2020	DNN	✗	✗	5G	RSSI	1.6	89%	160	simulations
[11]	2020	Collaborative localization	✗	✗	UWB	Range-free	<1m	-	-	simulations
[12]	2019	MLP and CNN	✗	✗	WiFi	RSSI et CSI	CNN-CSI : 0.92, MLP-RSS : 1.92	-	-	real
[14]	2019	CNN	✗	✗	WiFi	RSSI	-	94.13 %	$0.68 \cdot 10^{-3}$	simulations
[15]	2019	Trilateration	✗	✗	WiFi	RSSI	1.37	-	-	simulations
[16]	2018	Trilateration	✗	✗	UWB	ToF	0.12	-	-	real
[17]	2018	DNN	✗	✗	WiFi	CSI	0.6-0.7	93.9%-96%	-	real
[18]	2016	ANN	✓	PSO	WiFi	RSSI	1.893	-	43	real

example, in [24], a method was applied to take AoA measurements using LoRa (Long-Range Radio) signals.

- Time of flight (ToF): calculates the time required for a signal to travel from the source to the destination and vice versa. Based on this measurement and the speed of the radio signal propagation, the distance between two objects can be estimated. This solution is less affected by signal attenuation through obstacles or antenna directivity, compared to other measurement techniques. Ultra-Wide Band (UWB) signals allow precise Time of Flight (ToF) measurements, making UWB ToF measurements suitable for high-precision localization solutions. For example, in [25], a platform that can be used to develop and evaluate new methods proposed for indoor localization was presented. This platform was built based on UWB technology and the ToF measurement technique.
- Time Difference of Arrival (TDoA): is a variant of Time of Flight (ToF). It measures the difference between the arrival times of a signal from a single transmitter to two different receivers considered as two reference points. By combining this measurement with the signal transmission speed, the difference in the distances between the transmitter and the two receivers can be calculated [26].
- Range: it is an estimation of the distance between two objects obtained through a telemetry protocol. This measurement can be estimated from the received signal strength or by using Time of Flight (ToF).
- Signal-to-Noise Ratio (SNR) (expressed in decibels) [27]: is a measure that evaluates the quality of a transmission by comparing the level

of a desired signal to that of noise. It is defined as the ratio between the power of the desired output signal and the power of the noise.

Once these measurements are performed, localization algorithms will be implemented. Generally, they are based on triangulation techniques where the geometry of triangles is applied to calculate the locations of IoT objects. Triangulation can be classified into two techniques: (i) lateration, which is based on the distances to points with known locations, and (ii) angulation relying on the angles with respect to points having known locations in addition to distances. The taken measurements are used to estimate the positions of the objects. Trilateration is a technique that calculates the distances to at least three fixed references in a 2D environment. In a 3D environment, a minimum of four references is required. In fact, trilateration algorithms are the simplest and most common applied to localize the objects. Other possible localization solutions are those based on artificial intelligence techniques, specifically Machine Learning and Deep Learning techniques. These methods involve using data collected from the environment and utilizing appropriate algorithms to predict IoT object positions. The next section introduces the field of Deep Learning and presents its most common techniques.

3.2 Deep Learning

Machine Learning (ML) techniques are algorithms that enable computers to learn from data without being explicitly programmed. Deep Learning (DL), on the other hand, is a sub-field of Machine Learning that focuses on algorithms utilizing artificial neural networks inspired from the human brain to solve problems in several domains. In recent years, DL techniques have been successfully applied by many researchers and in various use cases.

Some of the DL techniques are presented below:

- Dense layers: They are the most commonly used layers in neural networks. As a dense layer is deeply connected to its previous layer, it receives inputs from all neurons in the previous layer.
- Deep Neural Network (DNN): It is composed of multiple layers including an input layer, an output layer and intermediate (hidden) layers. DNNs can handle nonlinear relationships and are capable of processing large amounts of data.
- Artificial Neural Network (ANN): They mimic the functioning of the human brain.
- Convolutional Neural Network (CNN): According to [28], CNN is the most popular DL architecture due to its efficiency in processing voluminous data. CNNs are widely employed in computer vision tasks.
- Recurrent Neural Network (RNN) [29]: In RNNs, connections between nodes can be recursive. In other words, the input of certain nodes depends on the previous output of the same nodes. This enables RNNs to process sequential data.
- Long Short-Term Memory (LSTM): LSTM, proposed in [30], is a type of recurrent neural network. It was designed to address the vanishing gradient problem and is largely utilized in tasks involving sequential data.
- Gated Recurrent Unit network (GRU): GRU is a variant of RNN characterized by a reduced number of parameters and fast convergence. It was introduced in [31].
- Multilayer Perceptron (MLP): MLP is a widely used type of neural network where connections are only made from input to output in the network (single direction). It includes no recurrence, meaning that the output of a neuron cannot affect its next input.
- Ensemble learning: This technique involves combining the outputs of two or more models to improve the obtained results. For example, some methods calculate the average of the outputs provided by multiple DL and ML models.

Indeed, any model among the previously-mentioned ones is trained by a learning process. In other words, a DL model has a set of trainable parameters whose values are modified and optimized during several iterations in order to minimize a loss function (error) at the training step. An iteration is called an epoch and the number of epochs must be predetermined together with the batch_size. Indeed, the training data is divided into batches (the batch_size refers to the number of samples in each batch). The weights of the model to be trained are typically initialized with random values. Then, at the end of processing each batch, these weights are

updated to minimize a loss function generally used to evaluate the difference between the actual values and the predicted values of the target outputs. In this process, an optimization algorithm is utilized. Among the most commonly-applied optimizers, we can mention:

- Gradient descent (GD): is a technique in ML and DL used to iteratively minimize a cost/loss function.
- Stochastic Gradient Descent (SGD): is a variant of GD. It is an optimization method that accelerates convergence and steers the weights in the right directions towards their optimal values [32]. SGD is one of the most popular optimization algorithms employed to train DL models due to its easy implementation and high efficiency [33].
- Batch Gradient Descent (BGD): is a variant of GD. It is an optimization algorithm where each iteration considers all training data [34].
- Follow the Regularized Leader (FTR): This algorithm was described in [35]. It is recommended to use it particularly with shallow models with a large number of features.

Another type of optimization algorithms is described in the following section. The latter introduces the meta-heuristic optimization and its most recent and widely used algorithms.

3.3 Meta-heuristic Optimization

The objective of an optimization problem is to find an optimal solution in a search space containing domains of possible values of decision variables. Such problem is resolved by reducing the search space to only the values that optimize a number of objective functions and satisfy a set of constraints. The problem including only one objective function is known as a single-objective optimization problem. Optimization problems are solved by meta-heuristic algorithms. The most recent and widely employed algorithms are depicted below:

- Grey Wolf Optimizer (GWO) [36]: is a meta-heuristic algorithm inspired from the social hierarchy and hunting behavior of grey wolves. [37] recommended using this algorithm to solve real-world optimization problems and proved its efficiency by describing various research works that use GWO and its different variants. For example, in [38], the multi-objective version of GWO (MOGWO) was utilized to model routing in an IoT environment. Experimental results showed the ability of MOGWO to overcome routing challenges in IoT data collection networks.
- Arithmetic Optimization Algorithm (AOA): is a meta-heuristic method based on the distribution behavior of the fundamental arithmetic operators in mathematics. Its performance was validated and

justified by the experimental results presented in [39].

- Whale Optimization Algorithm (WOA) [40]: is a nature-inspired meta-heuristic optimization algorithm relying on the behavior of whales and their hunting strategy. It is characterized by a reduced number of parameters, easy implementation and fast convergence. Therefore, it has gained wide popularity and has been applied to various problem domains. Authors, in [41], summarized the research works in which solutions based on WOA were proposed.
- Particle Swarm Optimization (PSO) [42]: is an optimization algorithm that emulates the social behavior of birds through algorithmic operations. It has been largely employed in several studies due to its simplicity and the reduced number of parameters it utilizes. Different variants of PSO and recent works that have used this algorithm were described in [43].

4 Methodology

The approach introduced in the current work consists in developing a Deep Learning model optimized by a meta-heuristic algorithm and trained on UWB Time of Flight data in order to estimate the positions of mobile objects in an IoT environment. This section presents first the developed approach. It introduces the proposed algorithms applied to model the training of the DL model using meta-heuristic optimization. Then, it describes the localization process based on the trained and optimized model.

4.1 Parameter Optimization with Meta-heuristic Algorithms

A Deep Learning model composed of dense layers is suggested and the learning of their weights is optimized using meta-heuristic algorithms. As mentioned in Subsection 3.2, during the training phase, multiple learning iterations are performed to find the values of the weights that help formulate the correct function mapping inputs x to outputs y . In other words, these weights are learned and improved during training. Equation 1 simplifies and generalizes the output equation of a layer utilized in a neural network.

$$y_j^t = f\left(\sum_i W_{ji}x_i^t\right) \quad (1)$$

where x_i^t and y_j^t represent, respectively, the input and output of a layer at time t . W_{ji} are the trainable weights during the learning phase. The values of the weights are updated by applying an optimizer algorithm in order to reduce the prediction error over the epochs.

The performance of the optimizer used during the training phase greatly affects the convergence speed of

the DL model and the quality of the predicted results it provides. Moreover, IoT applications require accurately-calculated positions and fast processing to leverage their full potential. To overcome these challenges, it is important to improve the performance of the DL model utilized for localization by applying an enhanced optimizer. Therefore and to deal with these issues, optimization meta-heuristics are used in the present work. By treating the DL model weight training as a meta-heuristic optimization problem and solving it with an algorithm among those dedicated in this paradigm, solutions can be generated and employed to update the weight values during different iterations.

Relying on this definition of the learning step, the weights of a DL model are optimized, in this study, by applying meta-heuristic algorithms. The weight learning is translated into a single-objective optimization problem. The objective function to minimize is the prediction error function and the variables, which constitute the population of the optimization problem, are the weights of the DL model. In other words, during the different iterations, the new values of the weights are the best values selected by the meta-heuristic algorithms (Figure 1).

The modeling of weight optimization with a meta-heuristic method is summarized in Figure 2 and detailed in Algorithms 1, 2 and 3.

At the beginning of the training phase, the weights were initialized, and a maximum number of epochs was defined. The initial weights were used to initialize the population of the meta-heuristic optimization algorithm. Then, a new iteration was executed until reaching the maximum number of epochs. During each epoch, all the training data were employed. These data were divided into batches. At the end of processing each batch, the loss function value was calculated. Then, it was proven whether the meta-heuristic optimization population had been fully constructed or updated with the latest iterations. In this case, the mathematical equations of the meta-heuristic optimization would be applied to calculate new weight values, and the algorithm's population was updated with the newly-computed weights. If it was not the case, the weights would be updated using the equations of Gradient Descent (GD). Once the maximum number of epochs was exceeded, the weights obtained at the last iteration would be considered optimal, showing the end of the training phase.

Algorithm 1 details the training phase of the developed model. First, the DL weights were initialized with random values together with the parameters of the meta-heuristic algorithm to be applied. Among these parameters, *pop_size*, the size of the population to be considered, was specified. Afterwards, a number of epochs were executed to train the model. Each epoch consists of a number of iterations equal to the size of the input data divided by the batch size to be taken into account. Algorithms 2 and 3 describe the processing performed during each iteration.

Figure 1 Weight optimization by applying a meta-heuristic algorithm

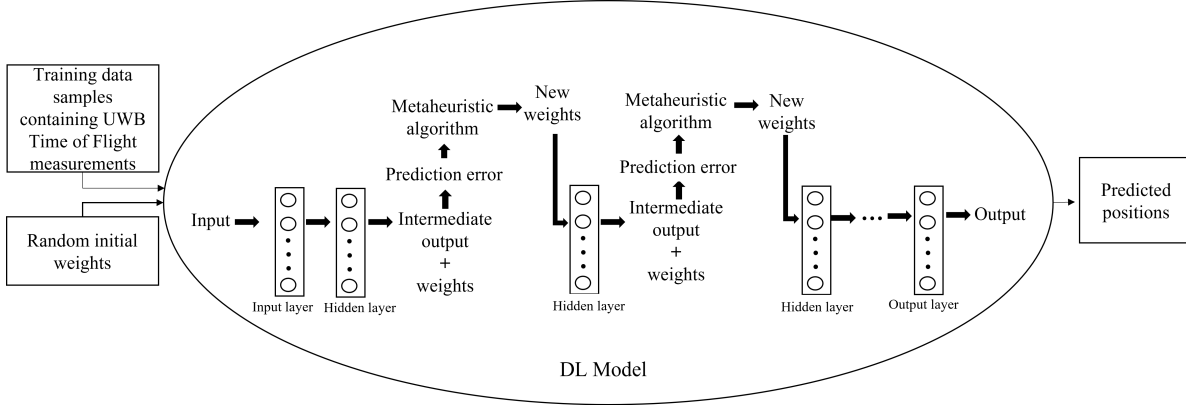
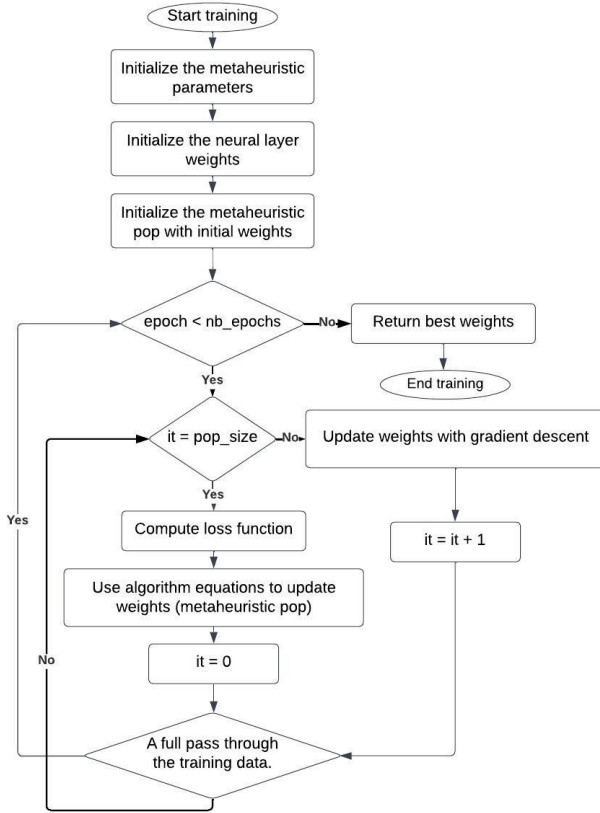


Figure 2 Flowchart of training neural layer weights by applying a meta-heuristic algorithm



Algorithm 1: train()

Input: $input_data$, pop_size , $nb_generations$, nb_epochs

Output: $parameters$

```

1  $weights \leftarrow$  random weights
2  $metaheuristic\_parameters \leftarrow$ 
   initializeMetaheuristicParameters()
3  $it \leftarrow 0$ 
4  $c\_pop\_write \leftarrow 0$ 
5  $c\_pop\_read \leftarrow 1$ 
6  $apply\_metaheuristic\_algo \leftarrow False$ 
7  $first\_pop \leftarrow True$ 
8  $pop \leftarrow \emptyset$ 
9 foreach  $epoch \in nb\_epochs$  do
10   foreach  $step \in nb\_iterations$  do
11     if  $apply\_metaheuristic\_algo$  is  $False$  then
12        $constructOrUsePop(input\_data, weights, pop,$ 
13          $first\_pop, pop\_size, c\_pop\_write,$ 
14          $c\_pop\_read, apply\_metaheuristic\_algo)$ 
15     else
16        $weights \leftarrow$ 
17          $trainWithMetaheuristicAlgo(pop,$ 
18            $nb\_generations, it,$ 
19            $metaheuristic\_parameters)$ 

```

During the various epochs, the meta-heuristic was applied after every pop_size iterations. In other words, the training phase was divided into stages of length pop_size iterations. During the first stage, the weight values were used to initialize the population. At the end of this stage, the meta-heuristic algorithm was applied to update the weights. During the subsequent stages, the weights generated by the previous stage were first used. At the end of each stage, the weights were optimized using the equations of the employed meta-heuristic algorithm.

4.2 Localization with the optimized DL model and Time of Flight (ToF)

The localization approach involves two phases (Figure 3):

Algorithm 2: constructOrUsePop()

Input: pop , $first_pop$, pop_size , c_pop_write ,
 c_pop_read , $apply_metaheuristic_algo$ **Output:** $parameters$

```

15 if  $first\_pop$  is True then
16    $parameters \leftarrow$ 
       $getParametersWithTraditionalMethod()$ 
17    $pop \leftarrow pop \cup parameters$ 
18    $c\_pop\_write \leftarrow c\_pop\_write + 1$ 
19   if  $c\_pop\_write = pop\_size$  then
20      $first\_pop \leftarrow False$ 
21      $apply\_metaheuristic\_algo \leftarrow True$ 
22 else
23    $parameters \leftarrow pop[c\_pop\_read]$ 
24    $c\_pop\_read \leftarrow c\_pop\_read + 1$ 
25   if  $c\_pop\_read > pop\_size$  then
26      $apply\_metaheuristic\_algo \leftarrow True$ 

```

Algorithm 3: trainWithMetaheuristicAlgo()

Input: pop , $nb_generations$,
 $apply_metaheuristic_algo$,
 $metaheuristic_parameters$ **Output:** *new pop*

```

27  $updatePopWithMetaheuristicAlgo(pop,$ 
       $metaheuristic\_parameters)$ 
28  $it \leftarrow it + 1$ 
29  $metaheuristic\_parameters \leftarrow$ 
       $updateParametersOfMetaheuristicAlgo(nb\_generations,$ 
       $it)$ 
30  $apply\_metaheuristic\_algo \leftarrow False$ 
31  $parameters \leftarrow pop[1]$ 
32  $c\_pop\_read \leftarrow 2$ 
33 return  $pop$ 

```

- Offline phase: The first step is to use the dataset presented in sub-section 5.1.1 to train the model. The data, primarily including UWB Time of Flight (ToF) measurements, serves as the input for the DL model to be trained. The coordinates of the mobile object represent the target results to be predicted, and the other columns are the data based on which the prediction will be made. The weights of the model were learned and optimized by applying meta-heuristic optimization algorithms. The objective function to minimize is the average distance between the actual positions and the estimated ones. During training, a portion of the data was utilized for validation. The result provided in this phase is a trained and optimized model.
- Online phase: The second phase represents a scenario where a mobile object needs to localize itself. In this case, after exchanging messages with the anchors, this object selects the necessary data (primarily including UWB ToF measurements) to be used for prediction. By giving these data to the model, resulting from the first phase, as input, the object can estimate its own position (coordinates).

5 Experimental Results and Discussion

In order to evaluate the performance of the developed and proposed model, a set of tests were conducted. This section presents first the experimental setup. Then, the obtained results are analyzed.

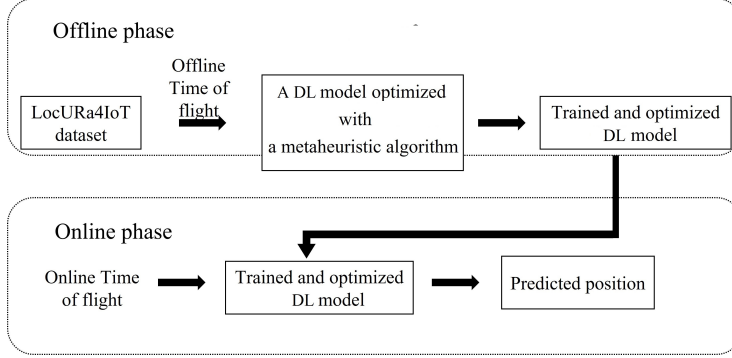
5.1 Experimental Setup

This section describes the experimental details including the used dataset, the employed evaluation criteria, the experimental parameters and the implementation choices.

5.1.1 The Dataset

The dataset utilized to apply the proposed solution was generated by the real-world platform LocURa4IoT [44] designed to develop and test approaches applied to solve the problem of indoor localization in IoT environments based on Time of Flight (ToF). Additionally, it can be employed to test and evaluate the ranging protocols in IoT networks.

The platform primarily uses Ultra-Wide Band (UWB) technology, which offers the highest precision and incorporates Bluetooth Low Energy (BLE) as well as LoRa transceivers. To construct the dataset, a set of anchors were installed in the considered IoT environment. A mobile object was moved within the environment to collect measurements regarding the signals exchanged between the object and the anchors.

Figure 3 Localization by applying an optimized DL model and using UWB Time of Flight

The Two-Way Ranging (TWR) protocol [45] was applied during the message exchange between the mobile object (TWR client) and the anchors (TWR servers). The dataset includes several measurements such as ToF, anchor positions, real positions of the mobile object, RSSI measurements, range measurements, etc. The ToF and range measurements were calculated using the TWR protocol [45]. The time difference between the TWR server clock and the TWR client clock can lead to inaccurate Range measurements. To deal with this problem, the Range measurements in the LocURa4IoT dataset were optimized. The dataset is available online at [46].

To apply the localization solution, the lines of the dataset were rearranged to separate the pairs of lines and avoid overfitting. Some data from the LocURa4IoT dataset were eliminated. They include:

- Columns with unique values such as object identifiers and the used protocol.
- Columns that do not provide useful information that can be relied on to localize objects.

Moreover, the dataset was reorganized such that each row would contain information about the interactions with the same 5 anchors. Then, the dataset was divided into three parts: for training, test and validation. Authors, in [47], considered the ratio ($p : \sqrt{p} : (\sqrt{p} + 1)$) as the optimal distribution ratio for the mentioned data parts. $p = \sqrt{N}$, N is the number of unique rows in the used data.

The new content of the dataset consists of 3947 unique rows. Therefore, $p = 63$ and the distribution ratio became $63 : 8 : 9$. In other words, the dataset was divided as follows:

- 78.75% of the data were used as training data
- 10% of the data were utilized as validation data
- 11.25% of the data were employed as test data

5.1.2 The evaluation criteria

Among the existing evaluation criteria, the two following criteria were used in the performed experiments:

- Mean Absolute Error (MAE): It calculates the average of the absolute differences between the actual target output values (the test data) and the predictions obtained by applying the model (Equation 2). A smaller MAE value indicates better prediction accuracy. As MAE is the most commonly-used criterion applied to evaluate the localization solutions, it is utilized in the present study.

$$MAE = 1/n \sum_i^n |y_i - \hat{y}_i| \quad (2)$$

y_i : actual value, \hat{y}_i : predicted value

- Mean Squared Error (MSE): It computes the squared difference between the actual target output values (the test data) and the predictions obtained by applying the model. Then, it takes the average of these errors (Equation 3). A smaller MSE value indicates better prediction accuracy. It represents the simplest and most common loss function. [48] recommended using this type of loss to solve prediction problems.

$$MSE = 1/n \sum_i^n (y_i - \hat{y}_i)^2 \quad (3)$$

y_i : actual value, \hat{y}_i : predicted value

5.1.3 Experimental Parameters

Table 2 contains the hyper-parameters utilized to configure the trained and tested DL models during the conducted experiments.

Table 2 Hyper-parameters of DL models

Hyper-parameter	Value
Number of epochs	2000
Batch size	32
Learning rate	0.001
Number of hidden layers	2

5.1.4 Implementation

The experiments were conducted on a computer equipped with an Intel i7 processor and 16 gigabytes of RAM. To implement the approach, the following choices were made:

- Python as the programming language
- Jupyterlab 3.5.1 IDE
- Keras [13], a Python library for Deep Learning. It provides a high-level Python interface with a high level of abstraction and offers simple and consistent high-level APIs, making it easier to develop and test Deep Learning models.
- Developing two optimizers to train the weights of a Deep Learning model. They are based on the following meta-heuristic algorithms:
 - GWO [36] : is utilized in the current work as it is one of the recent meta-heuristics that has recently gained significant interest in various domains [49].
 - AOA [39] : It was chosen because it is one of the most recent meta-heuristic methods proposed in 2021. Additionally, according to [39], it outperforms other well-known optimization algorithms and provides promising results in solving challenging optimization problems.

5.2 Results

In order to prove the good performance of the introduced approach, experiments were conducted, and the developed models were compared in terms of MAE and MSE. In addition, the best developed model was compared to a multi-lateration solution. This section presents the obtained results.

5.2.1 Comparison of the Optimization Methods

The training of 3 DL developed models was tested with the same hyper-parameters (Table 2), primarily based on the ToF measurements from the LocURa4IoT dataset. The used data includes anchor coordinates and ToF measurements. Two models were trained by applying the proposed method (Sub-section 4.1) and employing GWO and AOA, respectively. The third model was trained using the existing optimization method (GD).

Figure 4 shows the learning curves of the 3 employed models. It presents the variation of MAE values during the training epochs using the three optimizers. Each two curves were plotted together to facilitate the comparison of the models. It should be noted that, during the training epochs, the parameters were updated to find the optimal values that yielded lower MAEs. This figure demonstrates that optimizing the trainable parameters with the developed approach using GWO

Table 3 Comparison of training times

Optimization algorithm	Training time (seconds)
GWO	328.302
AOA	393.162
GD	281.52

resulted in the fastest convergence, compared to the other two optimizers. In other words, the MAE errors were smaller and decreased more rapidly when the GWO equations were used to update the weight values during the learning iterations. This figure also reveals that optimization employing the AOA meta-heuristic is less effective than that made utilizing GD.

Thus, we can conclude, from these curves, that our proposal applied with GWO outperforms the other two methods, followed by the GD optimizer. However, optimization with AOA is the least effective. This conclusion is further supported by Figure 5. In the latter, the average values of MAE and MSE errors across all epochs during the training phase are compared. This comparison confirms that training with GWO surpasses that done by the other two optimizers (GD and AOA) and that our proposal relying on GWO accelerates convergence towards the optimal parameters and enhances the learning of the DL model.

The learning times of the three models are compared. The provided results are presented in Table 3. In terms of the training time, optimization with GD is the fastest, followed by that with GWO and that with AOA. However, the learning time criterion is not crucial as it pertained to the off-line phase (training). Indeed, during the execution of IoT applications (on-line phase), objects used a pre-trained model to self-locate themselves.

Based on these results, the optimization relying on the proposed approach applied with GWO was considered to conduct the remaining tests presented in the rest of this section.

5.2.2 Comparison of the Measurement Techniques

The initial performed tests showed that training based on the developed approach applied with GWO yielded the best results. For this reason, this optimization method was utilized to train three DL models with the same hyper-parameters, but with different data selected from the measurements taken in LocURa4IoT. In addition to the anchor positions, the used data included, respectively, UWB Time of Flight (ToF), Received Signal Strength Indicator (RSSI) and Range measurements. The tests were performed during this stage to compare the performance of the three developed models in terms of the accuracy of position prediction. This comparison was made by training the three developed models and using them to predict positions employing the test data (11.25% of the dataset). The predicted x and y coordinates obtained from the tested models were also compared. Additionally, these predictions were considered to calculate the distance between the actual position of the mobile object and its estimated position. Figure 6 represents a comparative

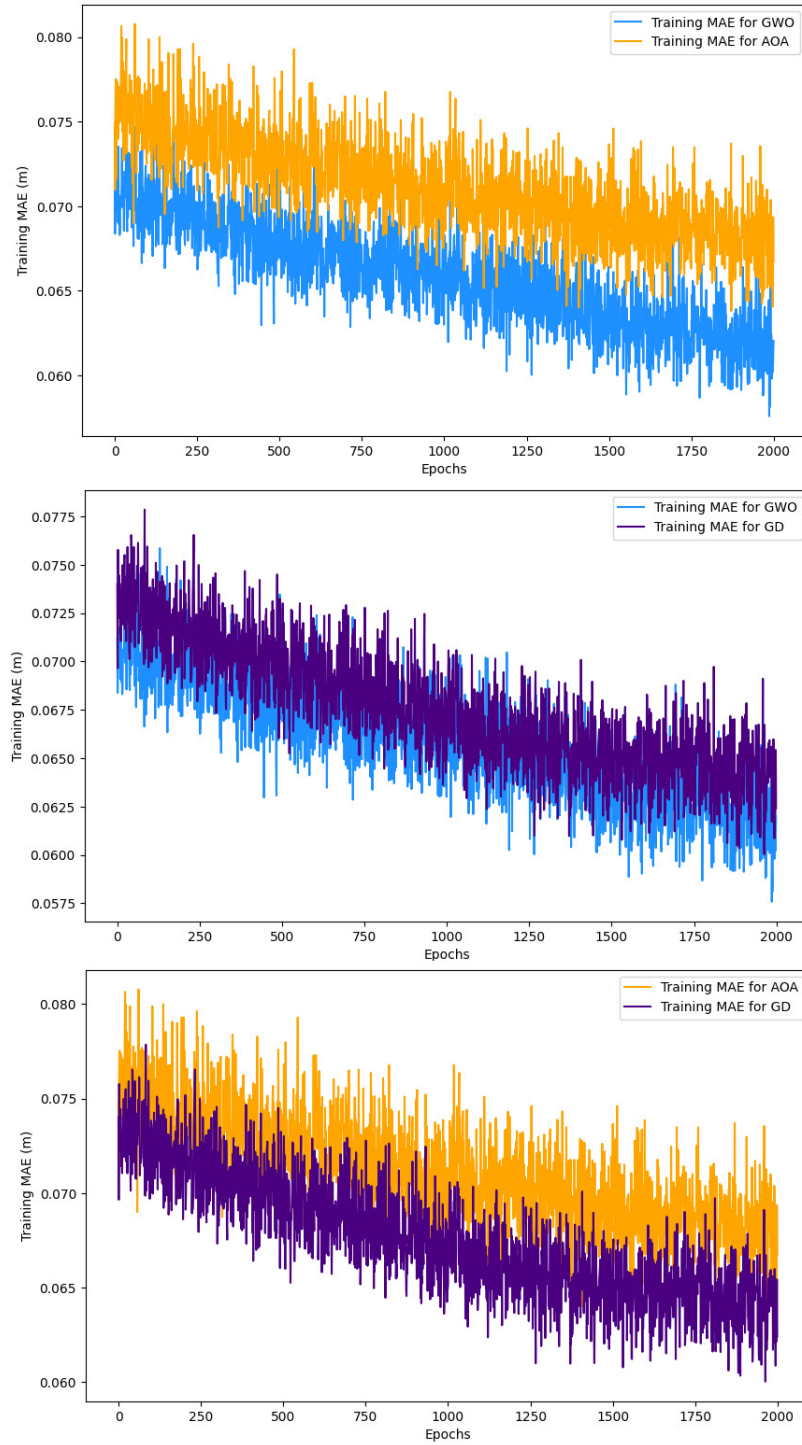
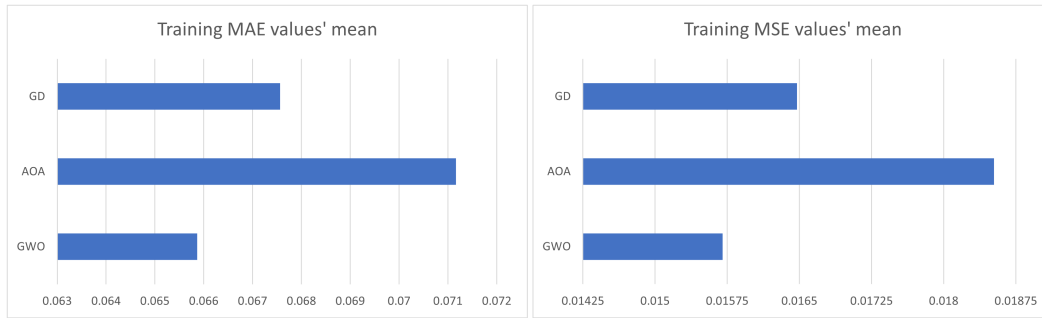
Figure 4 Learning curves for the DL models optimized with the proposed solution (employing GWO and AOA) and GD

Figure 5 Comparison of the overall average values of MAE and MSE errors across all training epochs



study of the results obtained by the three models. It displays the error variation curves obtained from the predictions given by applying the three tested models for x and y coordinates as well as the distance between the actual and the predicted positions. The curves indicate that the errors obtained from UWB ToF are generally lower than those provided by RSSI and Range measurements.

Figure 7 confirms that the positions predicted using UWB ToF measurements are more accurate than those obtained with RSSI and Range. Indeed, Figure 7 compares the Mean Absolute Error (MAE) and Mean Squared Error (MSE) values of the predictions obtained in terms of x and y coordinates and positions (distances between the actual and estimated positions). It is noticed that ToF measurements yielded the lowest values of both MAE and MSE criteria. It is also obvious that predictions with Range resulted in smaller MAE and MSE values than RSSI for the x coordinate and position. However, for the y coordinate, Range provided a lower MAE value and a slightly higher MSE value, compared to RSSI.

Based on these results, the optimization relying on the proposed approach applied with GWO and employing UWB ToF measurements was considered as the proposed indoor localization solution by this study, since it gives the best results compared to other developed and tested solutions. Therefore, it is compared to a conventional multi-lateration solution in the next sub-sub-section.

5.2.3 Comparison of the performance of the proposed solution with Multi-lateration Method

The objective of this test section is to demonstrate that the performance of the proposed approach is better than that of a conventional multi-lateration method presented in [50]. The position estimation capabilities of both solutions were tested using the same test data. The estimation results are compared in Figures 8 and 9.

Figure 8 compares the predicted values and the actual values of the x and y coordinates for the two compared solutions. The curves demonstrate that the predictions made by the introduced approach are more accurate than those of the multi-lateration method. Figure 9 presents

Figure 6 The error variation curves obtained from the predictions given by using ToF, RSSI and Range measurements in terms of x and y coordinates as well as the distance between real and predicted positions

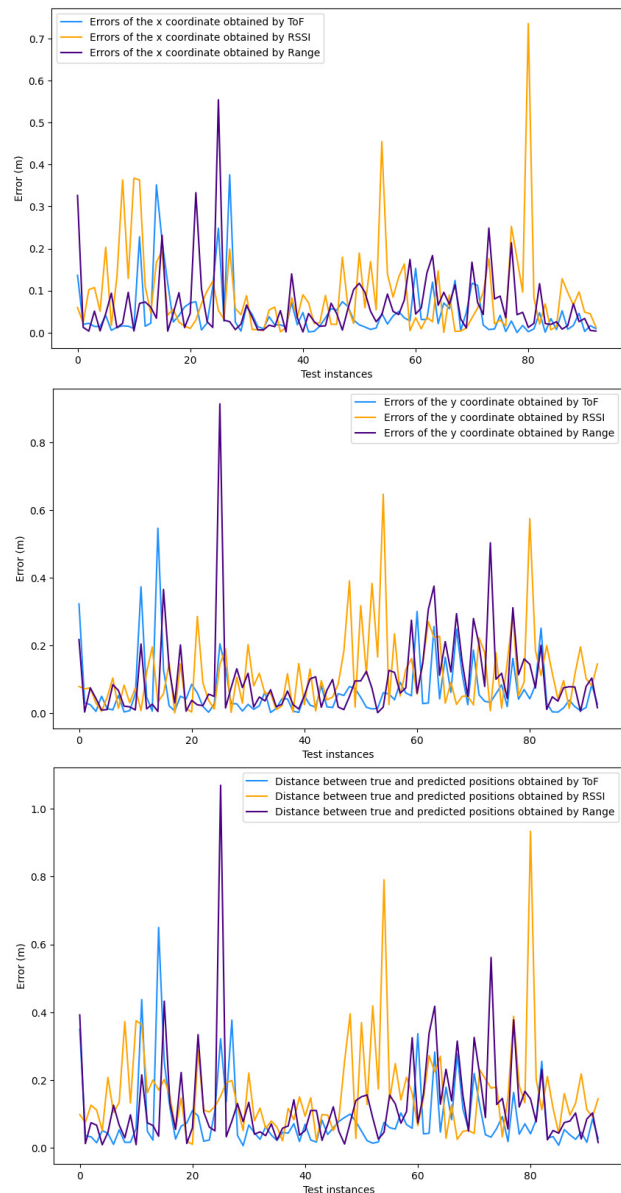


Figure 7 Comparison of the overall mean values of MAE and MSE errors for predictions of x and y coordinates as well as positions (distances between real and estimated positions)

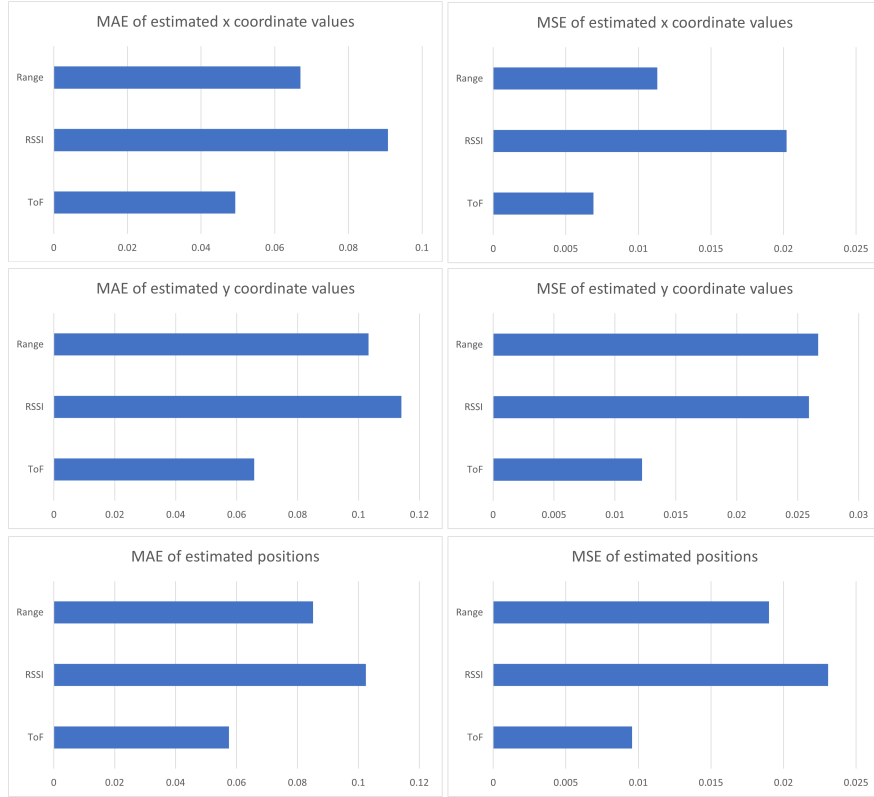


Figure 8 Comparison of the real coordinates and the estimated ones given by applying the suggested solution and the multilateration method

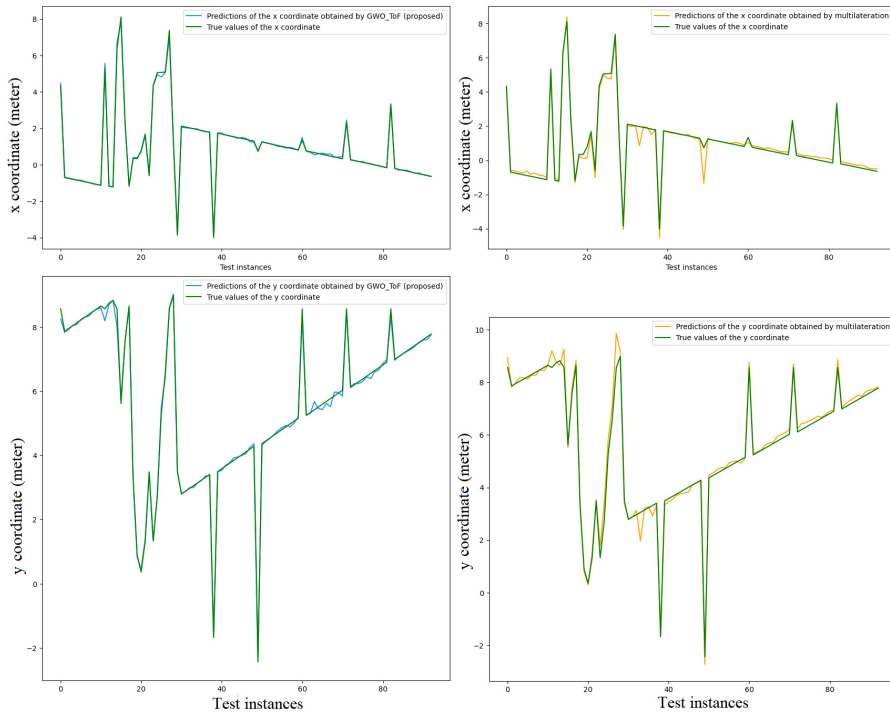
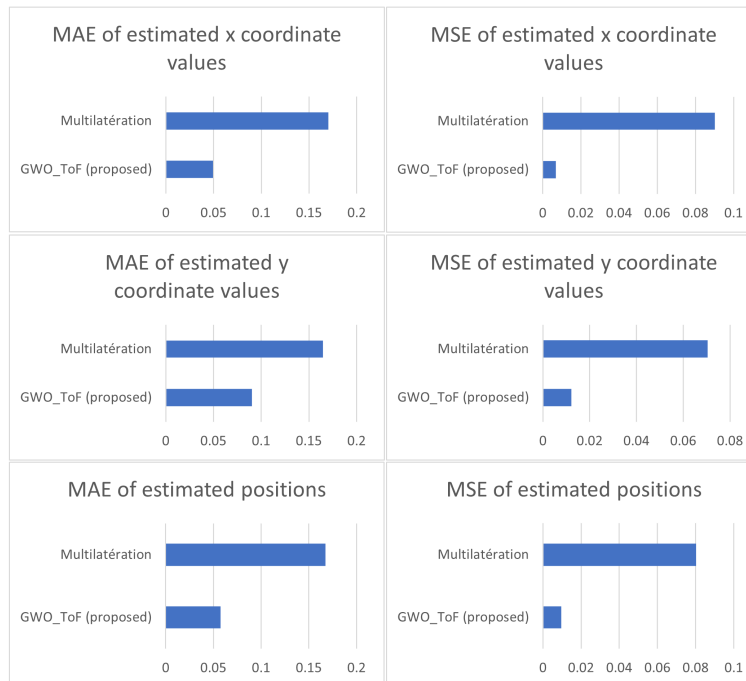


Figure 9 Comparison of the overall mean values of MAE and MSE errors for predictions of x and y coordinates as well as positions (distances between real and estimated positions)



a comparative study of the Mean Absolute Error (MAE) and Mean Squared Error (MSE) values calculated from the predictions of the coordinates and the distances between the actual and the estimated positions obtained by the two solutions.

These results reveal that the developed approach provided lower MAE and MSE errors for all three considered measurements (x, y and position). In fact, the rate at which the suggested localization solution achieved the minimum distance between the actual and the estimated positions is 93.54%.

Furthermore, the presented approach provided an average error between the actual and the predicted positions of 0.057m. These results also demonstrate that the proposed method surpassed the state-of-the-art solutions, described in Section 2, where the best average distance error was a few tens of centimeters.

The localization accuracy and the times required to apply the two localization solutions were also compared in table 4. The suggested solution outperformed the multi-lateration one by achieving a localization accuracy of 98.92%, whereas the latter achieved an accuracy of 89.24% (threshold = 0.5m). The time taken for position prediction using the introduced approach was 0.155 seconds, whereas the multi-lateration solution required 1.982 seconds to estimate the coordinates. Although the developed solution outperformed the other method in terms of the estimation time, it is worth to note that the proposed DL solution required an off-line step that necessitated 328.302 seconds to train the prediction model. On the other hand, the multi-lateration method did not require any additional phase apart from real-time position estimation.

5.2.4 Summary

Several conclusions can be drawn from the conducted tests and obtained results:

- Training the DL model parameters based on GWO, as proposed in this work, outperforms the other two optimization methods (AOA and GD). The approach applied with GWO accelerates convergence towards the optimal parameters and enhances the learning process for the DL model.
- Positions predicted using UWB ToF measurements are more accurate than those obtained with RSSI and Range.
- The introduced localization solution provided an average error between the actual and the predicted positions of 0.057m. It outperformed the tested multi-lateration solution that yielded an average distance error of 0.167m.
- The proposed technique surpassed the state-of-the-art solutions where the best results achieved an average distance error of few tens of centimeters.

6 Conclusion and Future Perspectives

In this article, a Deep Learning (DL) model trained with the Grey Wolf Optimization (GWO) algorithm and based on UWB Time of Flight (ToF) measurements was designed as a solution for indoor localization in an IoT environment. We conducted experiments using the LocURa4IoT dataset, constructed with measurements

Table 4 Comparison of the time required for applying the suggested solution and the multi-literation method

Solution	Localization accuracy	Training time (seconds)	Estimation time (seconds)
GWO-ToF (proposed)	98.92%	328.302	0.155
Multi-literation	89.24%	-	1.982

taken in a real IoT environment. A comparative study was conducted and presented to prove the high performance of the proposed model. Firstly, three Deep Learning models were trained employing the meta-heuristic optimization algorithms GWO and AOA as well as the GD algorithm, respectively. Their performance was evaluated during the training phase. It was demonstrated that GWO allowed better optimization of the trainable parameters by accelerating convergence towards the optimal values and outperformed the two other algorithms. Secondly, the capabilities of three DL models trained with GWO to predict positions based, respectively, on UWB ToF, RSSI and Range measurements were assessed. The obtained results showed that, compared to RSSI and Range, UWB ToF measurements provided more accurate positions, as indicated by lower Mean Absolute Error (MAE) and Mean Squared Error (MSE) values in the predictions. Thirdly, the predicted positions obtained using the suggested solution was compared with those calculated using an existing conventional multi-literation solution. The experimental findings proved that the introduced model outperforms the existing one in terms of MAE and MSE. It was also proven that the suggested solution achieved an average distance error of 0.057m, demonstrating its best superior performance, compared to state-of-the-art solutions where the best average distance error was a few tens of centimeters. Indeed, the proposed solution achieved a localization accuracy of 98.92%. However, the proposed solution took 328.302 seconds to train the DL model. As a continuation of this work, we plan to improve the training process to reduce the required learning time. Additionally, it will be beneficial to explore a dynamic solution that enables regular learning when the environment evolves over time. Furthermore, we will use the proposed localization solution to develop a geographic routing approach in a large-scale mesh IoT network.

References

- [1] Sandra Djosic, Igor Stojanovic, Milica Jovanovic, Tatjana Nikolic, and Goran Lj. Djordjevic. Fingerprinting-assisted uwb-based localization technique for complex indoor environments. *Expert Systems with Applications*, 167:114188, 2021.
- [2] Toni Perkovic, Lea Dujic Rodic, Josip Sabic, and Petar Solic. Machine learning approach towards lorawan indoor localization. *Electronics*, 12:457, 01 2023.
- [3] Tinh Pham and Ta Mai. Ensemble learning model for wifi indoor positioning systems. *IAES International Journal of Artificial Intelligence (IJ-AI)*, 10:200, 03 2021.
- [4] Xu Feng, Khuong An Nguyen, and Zhiyuan Luo. A survey of deep learning approaches for wifi-based indoor positioning. *Journal of Information and Telecommunication*, 6(2):163–216, 2022.
- [5] Danshi Sun, Erhu Wei, Zhuoxi Ma, Chenxi Wu, and Shiyi Xu. Optimized cnns to indoor localization through ble sensors using improved pso. *Sensors*, 21(6), 2021.
- [6] Quentin Vey, Francois Spies, Baptiste Pestourie, Denis Genon-Catalot, Adrien van den Bossche, Rejane Dalce, JULIEN SCHRIVE, and Thierry Val. Poucet: A multi-technology indoor positioning solution for firefighters and soldiers. 11 2021.
- [7] Beiya Yang, Erfu Yang, Leijian Yu, and Andrew Loeliger. High-precision uwb-based localisation for uav in extremely confined environments. *IEEE Sensors Journal*, 22(1):1020–1029, 2022.
- [8] Liping Wang and Sudeep Pasricha. A framework for csi-based indoor localization with 1d convolutional neural networks, 2022.
- [9] Guannan Liu, Hsiao-Chun Wu, Weidong Xiang, Jinwei Ye, Yiyang Wu, and Limeng Pu. Indoor object localization and tracking using deep learning over received signal strength. In *2020 IEEE International Symposium on Broadband Multimedia Systems and Broadcasting (BMSB)*, pages 1–6, 2020.
- [10] Brahim El Boudani, Loizos Kanaris, Akis Kokkinis, Michalis Kyriacou, Christos Chrysoulas, Stavros Stavrou, and Tasos Dagiuklas. Implementing deep learning techniques in 5g iot networks for 3d indoor positioning: Delta (deep learning-based co-operative architecture). *Sensors*, 20, 09 2020.
- [11] Quentin Vey, Rejane Dalce, Adrien van den Bossche, and Thierry Val. A distributed algorithm for range-based localization in sparse wireless networks. In *2020 30th International Telecommunication Networks and Applications Conference (ITNAC)*, pages 1–8, 2020.
- [12] Chaur-Heh Hsieh, Jen-Yang Chen, and Bo-Hong Nien. Deep learning-based indoor localization using received signal strength and channel state information. *IEEE Access*, 7:33256–33267, 2019.
- [13] Francois Chollet et al. Keras, 2015.

- [14] Wafa Njima, Iness Ahriz, Rafik Zayani, M. Terre, and Ridha Bouallegue. Deep cnn for indoor localization in iot-sensor systems. *Sensors*, 19:3127, 07 2019.
- [15] Naren Tada, Tejas Patalia, and Shivam Trivedi. Implementation of trilateration based localization approach for tree monitoring in iot. In Navin Kumar and R. Venkatesha Prasad, editors, *Ubiquitous Communications and Network Computing*, pages 115–123, Cham, 2019. Springer International Publishing.
- [16] Francois Despaux, Adrien van den Bossche, Katia Jaffres-Runser, and Thierry Val. N-tw: An accurate time-of-flight-based n-ary ranging protocol for ultra-wide band. *Ad Hoc Networks*, 79:1–19, 2018.
- [17] Tao Li, Hai Wang, Yuan Shao, and Qiang Niu. Channel state information-based multi-level fingerprinting for indoor localization with deep learning. *International Journal of Distributed Sensor Networks*, 14:155014771880671, 10 2018.
- [18] Nan Li, Jiabin Chen, Yan Yuan, Xiaochun Tian, Yongqiang Han, and Mingzhe Xia. A wi-fi indoor localization strategy using particle swarm optimization based artificial neural networks. *International Journal of Distributed Sensor Networks*, 12(3):4583147, 2016.
- [19] N. Bulusu, J. Heidemann, and D. Estrin. Gps-less low-cost outdoor localization for very small devices. *IEEE Personal Communications*, 7(5):28–34, 2000.
- [20] Drago Niculescu and Badri Nath. Dv based positioning in ad hoc networks. *Telecommunication Systems*, 22:267–280, 01 2003.
- [21] Trifun Savic, Xavier Vilajosana, and Thomas Watteyne. Constrained localization: A survey. *IEEE Access*, 10:1–1, 05 2022.
- [22] Satish Jondhale, Maheswar Rajagopal, and Jaime Lloret. *Survey of Existing RSSI-Based LT Systems*, pages 49–64. 01 2022.
- [23] Mohammed A. A. Al-qaness, Mohamed Elsayed Abd Elaziz, Ahmed Ewees, Aaqif Abbasi, Yousif Alhaj, and Ammar Hawbani. Channel state information from pure communication to sense and track human motion: A survey. *Sensors*, 19:3329, 07 2019.
- [24] Hussein Zeaiter, Francois Spies, Oumaya Baala, and Thierry Val. Measuring accurate angle of arrival of weak lora signals for indoor positioning. In *2022 IEEE 12th International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–8, 2022.
- [25] Quentin Vey, Rejane Dalce, Adrien Van Den Bossche, and Thierry Val. Indoor uwb localisation: Locura4iot testbed and dataset presentation. In *2022 IEEE 47th Conference on Local Computer Networks (LCN)*, pages 258–260, 2022.
- [26] Ashish Garg and Amit Gupta. Indoor tracking using ble -brief survey of techniques. 03 2020.
- [27] Toan Dinh, Thanh Nguyen, Hoang-Phuong Phan, Van Dau, Dzung Dao, and Nam-Trung Nguyen. Physical sensors: Thermal sensors. In Roger Narayan, editor, *Encyclopedia of Sensors and Biosensors (First Edition)*, pages 20–33. Elsevier, Oxford, first edition edition, 2023.
- [28] Madhusmita Sahu and Rasmita Dash. *A Survey on Deep Learning: Convolution Neural Network (CNN)*, pages 317–325. 01 2021.
- [29] Ke-Lin Du and M.N.s Swamy. *Recurrent Neural Networks*, pages 337–353. 12 2014.
- [30] Sepp Hochreiter and Jurgen Schmidhuber. Long short-term memory. *Neural computation*, 9:1735–80, 12 1997.
- [31] Kyunghyun Cho, Bart van Merriënboer, Dzmitry Bahdanau, and Yoshua Bengio. On the properties of neural machine translation: Encoder-decoder approaches, 2014.
- [32] I. Sutskever, J. Martens, G. Dahl, and G. Hinton. On the importance of initialization and momentum in deep learning. *30th International Conference on Machine Learning, ICML 2013*, pages 1139–1147, 01 2013.
- [33] J. Anila Sharon, A. Hepzibah Christinal, D. Abraham Chandy, and Chandrajit Bajaj. Chapter 11 - application of intelligent edge computing and machine learning algorithms in mbti personality prediction. In D. Jude Hemanth, Brij B. Gupta, Mohamed Elhoseny, and Swati Vijay Shinde, editors, *Intelligent Edge Computing for Cyber Physical Applications*, Intelligent Data-Centric Systems, pages 187–215. Academic Press, 2023.
- [34] Kshitij Tiwari and Nak Young Chong. 5 - preliminaries: A primer. In Kshitij Tiwari and Nak Young Chong, editors, *Multi-robot Exploration for Environmental Monitoring*, pages 41–52. Academic Press, 2020.
- [35] H. McMahan, Gary Holt, D. Sculley, Michael Young, Dietmar Ebner, Julian Grady, Lan Nie, Todd Phillips, Eugene Davydov, Daniel Golovin, Sharat Chikkerur, Dan Liu, Martin Wattenberg, Arnar Hrafnelsson, Tom Boulos, and Jeremy Kubica. Ad click prediction: a view from the trenches. pages 1222–1230, 08 2013.

- [36] Seyedali Mirjalili, Seyed Mirjalili, and Andrew Lewis. Grey wolf optimizer. *Advances in Engineering Software*, 69:46 – 61, 03 2014.
- [37] Isha Sharma, Vijay Chahar, and Sanjeevani Agri. A comprehensive survey on grey wolf optimization. *Recent Patents on Computer Science*, 09 2020.
- [38] Sihem Tlili, Sami Mnasri, and Thierry Val. A multi-objective gray wolf algorithm for routing in iot collection networks with real experiments. In *2021 National Computing Colleges Conference (NCCC)*, pages 1–5, 2021.
- [39] Laith Abualigah, Ali Diabat, Seyedali Mirjalili, Mohamed Elsayed Abd Elaziz, and Amir Gandomi. The arithmetic optimization algorithm. *Computer Methods in Applied Mechanics and Engineering*, 376:113609, 04 2021.
- [40] Seyedali Mirjalili and Andrew Lewis. The whale optimization algorithm. *Advances in Engineering Software*, 95:51–67, 05 2016.
- [41] Farhad Soleimanian Gharehchopogh and Hojjat Gholizadeh. A comprehensive survey: Whale optimization algorithm and its applications. *Swarm and Evolutionary Computation*, 48:1–24, 2019.
- [42] Leticia Cagnina, S. Esquivel, and Carlos Coello. A particle swarm optimizer for multi-objective optimization. *JCST*, 5, 12 2005.
- [43] Tareq M. Shami, Ayman A. El-Saleh, Mohammed Alswaiti, Qasem Al-Tashi, Mhd Amen Summakieh, and Seyedali Mirjalili. Particle swarm optimization: A comprehensive survey. *IEEE Access*, 10:10031–10061, 2022.
- [44] Adrien van den Bossche, Rejane Dalce, and Thierry Val. locura4iot-a testbed dedicated to accurate localization of wireless nodes in the iot. *IEEE Sensors Journal*, 12 2021.
- [45] Francois Despaux, Adrien van den Bossche, Katia Jaffres-Runser, and Thierry Val. N-twr: An accurate time-of-flight-based n-ary ranging protocol for ultra-wide band. *Ad Hoc Networks*, 79, 05 2018.
- [46] Locura4iot dataset. <http://locura4iot.irit.fr/datasets/20220611-uw-b-twr/>. Accessed: 2023-02-20.
- [47] V. Roshan Joseph. Optimal ratio for data splitting. *Statistical Analysis and Data Mining: The ASA Data Science Journal*, 15(4):531–538, 2022.
- [48] Qi Wang, Yue Ma, Kun Zhao, and Yingjie Tian. A comprehensive survey of loss functions in machine learning. *Annals of Data Science*, 9, 04 2022.
- [49] Emmanuel Dada, Stephen Joseph, David Oyewola, Alaba Fadele, Haruna Chiroma, and Shafi i Abdulhamid. Application of grey wolf optimization algorithm: Recent trends, issues, and possible horizons. *GAZI UNIVERSITY JOURNAL OF SCIENCE*, 35:485–504, 03 2022.
- [50] Abdelmoumen Norrdine. An algebraic solution to the multilateration problem. 04 2015.
- [51] Kyunghyun Cho, Bart Merriënboer, Dzmitry Bahdanau, and Y. Bengio. On the properties of neural machine translation: Encoder-decoder approaches. 09 2014.
- [52] M. El-Shorbagy and Aboul Ella Hassanien. Particle swarm optimization from theory to applications. *International Journal of Rough Sets and Data Analysis*, 5, 01 2018.
- [53] Chitra Dhawale, Kritika Dhawale, and Rajesh Dubey. *A Review on Deep Learning Applications*, pages 21–31. 01 2020.
- [54] Hassan Ramchoun, Mohammed Amine, Janati Idrissi, Youssef Ghanou, and Mohamed Ettaouil. Multilayer perceptron: Architecture optimization and training. *International Journal of Interactive Multimedia and Artificial Intelligence*, 4:26–30, 01 2016.
- [55] Diederik P. Kingma and Jimmy Ba. Adam: A method for stochastic optimization, 2014.
- [56] Boris Ginsburg, Patrice Castonguay, Oleksii Hrinchuk, Oleksii Kuchaiev, Vitaly Lavrukhin, Ryan Leary, Jason Li, Huyen Nguyen, Yang Zhang, and Jonathan M Cohen. Training deep networks with stochastic gradient normalized by layerwise adaptive second moments. 2019.
- [57] Adrien Van den Bossche, Rejane Dalce, Nicolas Gonzalez, and Thierry Val. Locura: A new localisation and uwb-based ranging testbed for the internet of things. In *2018 International Conference on Indoor Positioning and Indoor Navigation (IPIN)*, pages 1–6, 2018.