Neural Network Models for Calculation of Angle-of-Arrival in Indoor Positioning Systems

João Manuel Miranda Costa
Faculty of Engineering, University of Porto, Portugal

June 2022

Contents

List of Figures

Abstract

As industries and manufactures chase a more automated production process, real time asset localization research have been increasing. Since real environments embody a significant quantity of noise, more robust methods are required in order to achieve more accurate angle-of-arrival estimates. In this article three different artificial neural network models are tested addressing the viability of these in angle-of-arrival inference. The results show that higher hidden layer density models handle noisy environment effects better, as this ability decline as noise grows.

1 Introduction

The current interest towards real-time localization of devices or assets in context-specific indoor spaces has led to a vast number of studies related to the topic. One of the approaches took advantage of the direction finding capabilities that were recently added within the Bluetooth 5.1 version. Briefly, the direction finding is performed by sampling a Constant Tone Extension (CTE) which is appended to the end of a Bluetooth packet, in which a Radio Frequency (RF) device with an array of antennas can sample the phase of this tone at each of the antennas, thus deriving the angle-of-arrival (AoA) of the signal via the phase differences between antennas. In real world scenarios the raw phase measurements won't be the most accurate [1], hence disfiguring the phase difference profile, i.e, the graphical curve made up by eight data points referring each phase difference of each pair of antennas. Since unreliable AoAs are derived from this distorted profile, more robust methods are required to refine this process. Since the last few years, as Neural Networks hold on to, progressively, express more accurate outcomes, these algorithms have been applied to a vast range of problems that have been thought impractical, such as language translation, autonomous driving, face recognition, image generation, and so on. In this paper, it is presented the viability of various Neural Network models on inferring AoAs from several levels of noise on phase difference profiles, aiming to conclude if these structures are capable to reach the same achieving levels of the mentioned before, but for indoor localization.

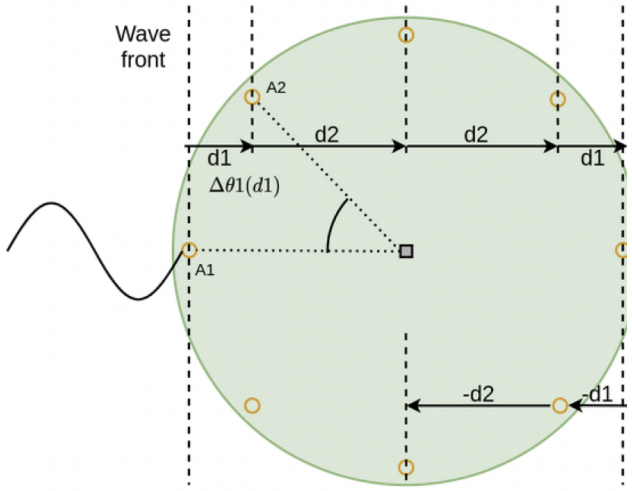

Figure 1: Model of circular antenna array, and relationship between angle, and travelled distance between antenna pairs

1.1 System Architecture

As Figure 1 demonstrates, the physical hardware structure developed in [1] is a circular printed circuit board equipped with a single Nordic Semiconductor nRF52811 micro-controller. This board contains all eight antennas equally spaced, i.e., placed at 45° steps relative to the center of the board.

The above figure also illustrates the wave front propagation for a certain AoA. The antennas are sequentially sampled in a clockwise order, so the first phase difference to be calculated is between antennas A1 and A2. Since the micro-controller only allows the process of sampling a single antenna in a given time instant, in [1] the phase difference, for example, between A1 and A2, is calculated based on the sampled results in A1, which are then used to predict the phase value that would be sampled at the moment that the sampling process in A2 is occurring.

This prediction, illustrated in Figure 2, is possible from the fact that the wave phase grows in a monotonous and linear way.

So, if we were to calculate the AoA purely through naive analytical methods utilizing a much simpler architecture, say a linear array containing only two antennas, as shown in Figure 3, the phase difference would immediately give the AoA. The problem with this intuitive approach lies in the fact that real environments are noisy, thus resulting in a wide range of phase difference

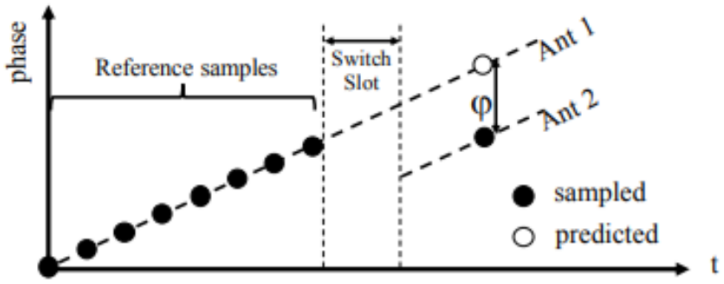

Figure 2: Phase difference prediction

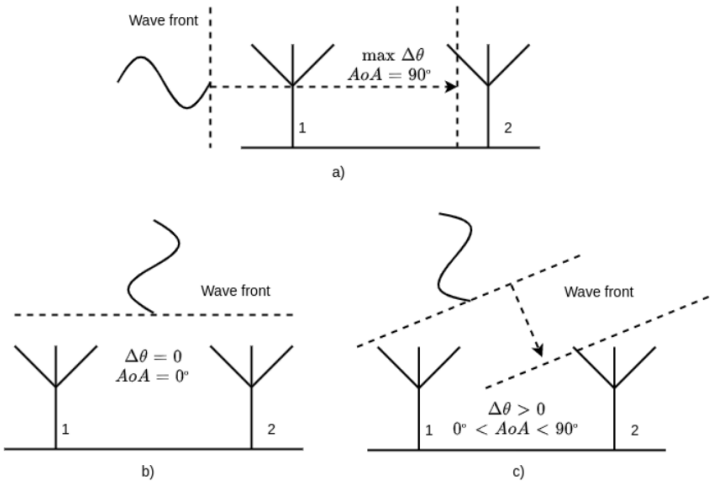

Figure 3: Wave incidence on a pair of antennas covering three different angles of arrival (AoA)

profiles characterizing the same AoA.

Figure 4 shows the wave phase on two antennas if simultaneous sampling was executed. Since the boards microcontroller can only sample a single antenna at a given moment, the obtained data, as illustrated in Figure 5, is a superposition of the curves in Figure 4. A tensor containing the phase differences relative to each pair of antennas can then be derived from the second graph data.

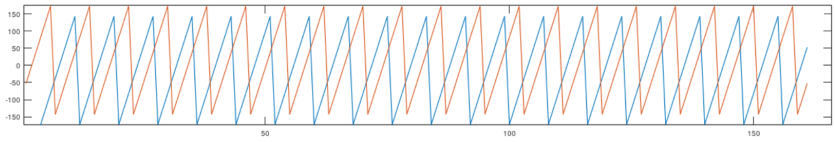

Figure 4: Sampled raw phases of two antennas

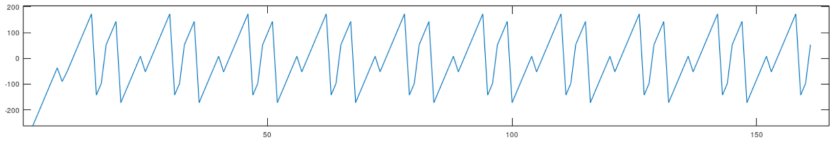

Figure 5: Microcontroller sampled raw phases

1.2 Data Interpretation

To get a better understanding on the data that has been captured by the circular board, the same data that, after being processed (from raw wave phases to phase difference profiles), is used to train the various Neural Network models, the picture below, Figure 6, shows the phase difference profiles, under a neutral value of noise, of two different waves, each corresponding to a different AoA. Because of the symmetry constituting the circular board containing all the antennas, these being equally spaced, these profiles illustrate themselves as sinusoidal curves. Figure 7 presents the phase difference profiles, of a given AoA, and it's a crystal clear example on how environmental noise (eg.: multiple indoor building reflections) can affect these profiles. Since each profile describes the same AoA, it's very difficult for naive analytical methods to handle such profiles distortion, hence the need for more robust methods.

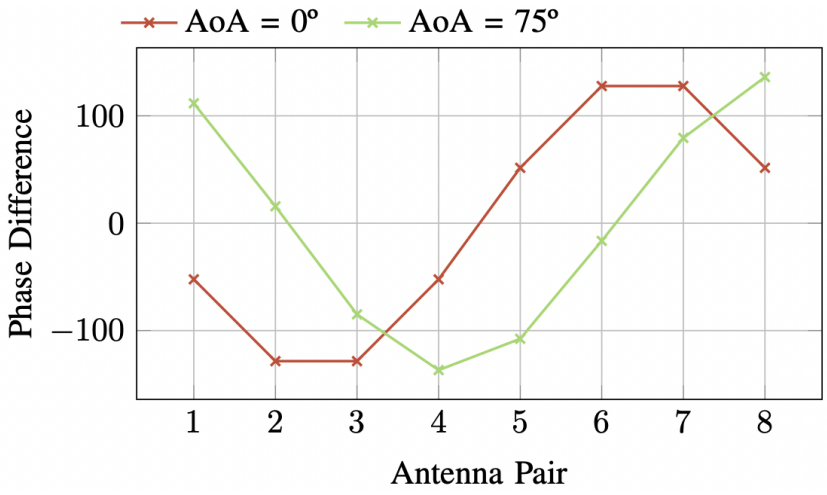

Figure 6: Profile of calculated phase difference values for all adjacent antenna pairs, as a function of two different AoAs, using phase sample data generated according to the models [1]

2 Related Work

2.1 Background

The approach taken to calculate an object coordinates inside a facility (Indoor Localization), requires the prediction of the angle of arrival (AoA) from the phase difference profile. Essentially, the AoA is the direction from which the signal is received by the array of antennas. The prediction of the AoA is made by an, previously trained, Artificial Neural Network. This latter mimicks the way that biological neurons signal to one another resulting in a computing system or algorithm capable of discover hidden patterns and features on raw datasets. To ease the development and assessment of this systems, the framework known as Tensorflow, devoloped by Google, was used. TensorFlow is an end-to-end open source platform for machine learning. It has a comprehensive, flexible ecosystem of tools, libraries and community resources that lets researchers push the state-of-the-art in machine learning and developers easily build and deploy ML powered applications.

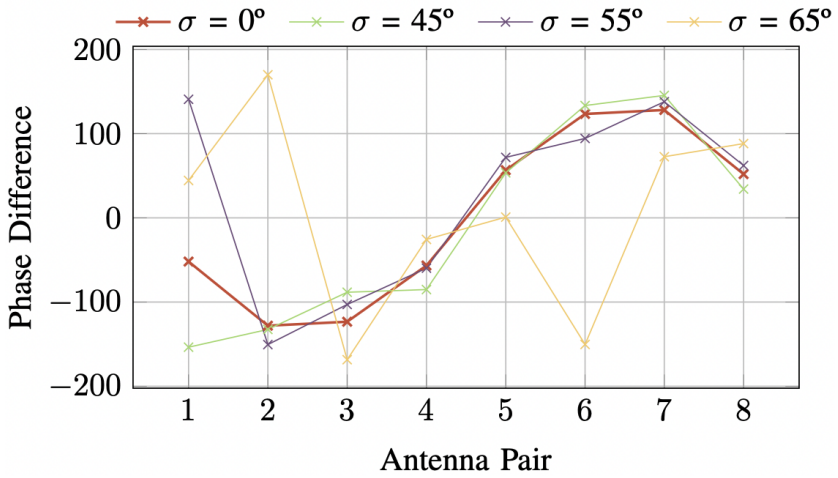

Figure 7: Observable effect of noise levels over the waves phase difference profile [1]

2.2 State of the art

In recent years, in order to both accurately and efficiently solve the AoA estimation problem, many ML approaches have been enforced, having the development of a regression model as it core idea. In some early studies on this topic, [2] modeled AoA estimation as a non-linear mapping problem and used radial basis function (RBF) neural network to solve it. The results show that the RBF neural network reduced the time consumption for AoA estimation computation. In [3], the developed angle estimation method is based on the use of a support vector regression (SVR) approach for the approximation of the unknown mapping that performs the transformation from the outputs of antenna array to angle information. In [4], a signal processing and machine learning combined tool is proposed for AoA estimation. The results show that the proposed approach for AoA estimation provides an accuracy and performance improvement compared with the Multiple Signal Classification (MUSIC) algorithm. Similarly, [5] proposed a learning-based scheme for achieving super-resolution AoA estimation and channel estimation in the massive multiple-input multiple-output (MIMO) system. Simulation results demonstrate that the proposed scheme can achieve better performance in terms of AoA estimation.

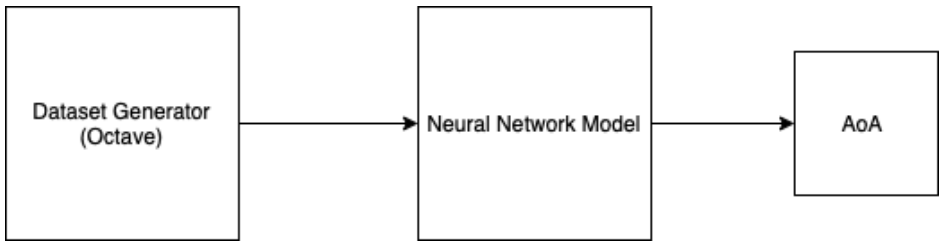

Figure 8: Flow Diagram

3 Proposed Approach

To refine the process of AoA inferring from various distortion levels on phase difference profiles, three Neural Network models were trained and tested with the intention of understanding the feasibility of this algorithm to improve not only accuracy, but performance as this systems demand real-time response.

3.1 Synthetic Processing Diagram

The processing diagram which describes the main phases that take raw phase measurements and turn these into inferred AoA's contains two primary components. The first phase refers to data generation which is executed by running a Octave script (code can be found available on GitHub)[1]. . Essentially, the script simulates the process developed in [1] returning a dataset containing an arbitrary quantity of data points, given the desired angle and noise value. Having the data collected, the next step is to use it to train and test the developed models. These can then be used to infer a AoA of any possible value ([0..360]) given a phase difference profile passed as its input.

[1] `https://github.com/JJJJJJJJJJJJJJl/NNsIndoorPosition/blob/main/dataset\_generator/generation/extract.m`

3.2 Neural Network Models

Model 1 The first model characterizes a very simple Neural Network composed by only 1 hidden layer containing a total of 128 neurons. The purpose of this model is to serve as a control model for later comparisons. The Python code[2] constitutes the first model described previously. As noticeable, Tensorflow eases the creation of a model by subclassing the Model class. In the constructor, the model receives the parameters needed and the layers are defined through the tf.keras.layers module. Here, two layers were defined, the first layer corresponds to the hidden layer with 128 neurons each using a relu activation funtion, then, the output layer with a single neuron employing a linear activation function.

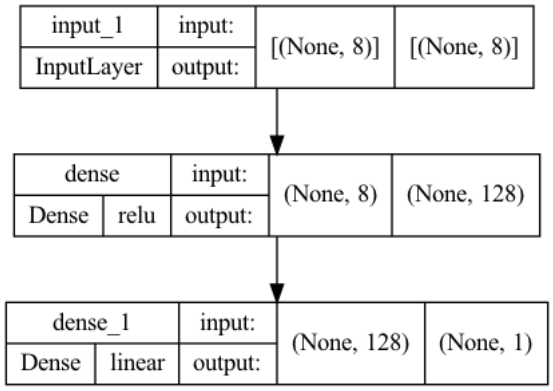

Figure 9: Model 1 visual representation

Model 2 The second model constitutes an heavier version of the first model. It contains three hidden layer with 128, 512 and 256 neurons respectively. Similarly to the previous one, but in this case, 3 hidden layers were created each receiving the number of hidden nodes from the constructor parameters. The choice of getting the number of neurons of each layer from the constructor parameters was matter of generalization.

[2] `https://github.com/JJJJJJJJJJJJJJl/NNsIndoorPosition/blob/main/NeuralNetwork.py`

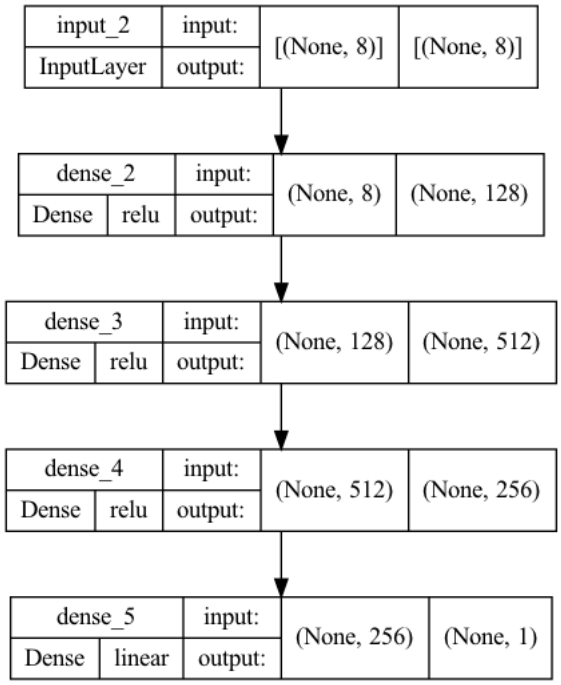

Figure 10: Model 2 visual representation

Model 3 The last model, a convolutional Neural Network, was designed and tested in order to assess the feasibility of this kind of network. It is composed of three convolutional layers followed by the same three hidden layer structure applied on the second model. To construct this kind of model, a first reshape layer was used to reshape the input from (8,), i.e, a 1 dimensional tensor of size 8, to (8,1) meaning 8 vectors of size 1 each, as the next layer, a Conv1D layer, demands it, followed by a MaxPooling1D layer for most-important data features extraction. After two more identical convolutional layer pairs (Conv1D and MaxPooling1D), a flatten layer is used to convert the current (2,64) shape of the data into a 1-Dimensional tensor of size 128. Finally, the same layer structure developed in the second model is appended next to the flatten layer (these are not explicitly visible in the model visual representation by means of brevity).

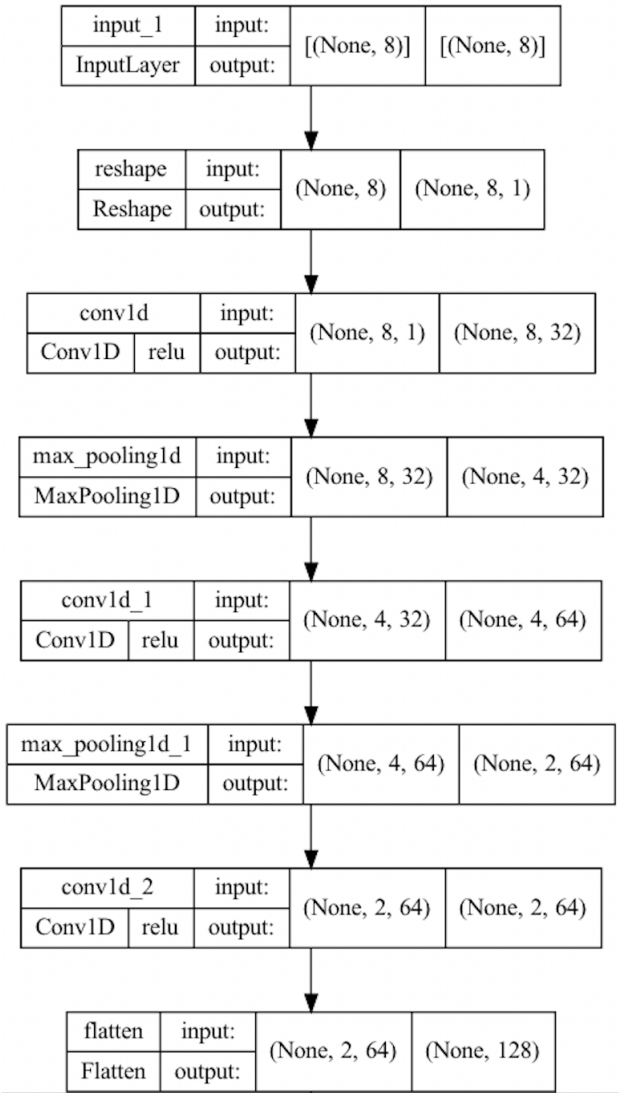

Figure 11: Model 3 visual representation

4 Experimental Evaluation

In order to experimentally assess the models performance, it is first needed to generate the datasets using the below explained script. Following data collection and preprocessing, the models are trained on the training dataset to then be tested against the testing dataset, both datasets generated in identical conditions.

4.1 Dataset Generator

All the data utilized for training and testing the models was derived from a set of Octave scripts. These allow the generation of phase difference profiles with an arbitrary amount of noise, followed by the corresponding AoA. Essentially, a single data point generated by the script is a 1-Dimensional tensor with nine columns, where the first eight constitute the eight phase differences from each adjacent antennas pair, and the last column is the label, i.e, the AoA matching the generated phase difference profile.

4.2 Experimental Setup

In order to obtain consistent results, all three models were trained and tested under the same exact datasets. The noise induced on the datasets range from a value of 10 to 90, with 10 units increments. So, nine training datasets, each with 72000 datapoints (200 datapoints dedicated to all possible angle natural values [0^{o}..360^{o}]), were generated, the same applies to the testing datasets. To assess the models performance two metrics were tracked, the Mean Squared Error, as the loss function, plus the Mean Absolute Error. Also, before feeding the data to the models, a preprocessing step consisting in data normalization, i.e, reducing data range values from [-180^{o}..180^{o}] (phase differences - features) and [0^{o}..360^{o}] (AoA - label) into [0..1] interval, was applied for training optimization inducements.

4.3 Results and Analysis

After testing each model on all test datasets, the results were plotted simultaneously for model comparison purposes. The obtained Mean Absolute Error graph shows an equivalent performance on models 2 and 3, slightly higher compared to Model 1. In all three models, the mean error exponentially

increases as phase difference distortion becomes more relevant. It is, still, worth noting the relevant AoA estimation accuracy, with a mean absolute error of $\leq$10 degrees, up to a noise level of 50. Moving to models temporal metrics, Table 1 presents the obtained training and testing times relative to all models. As expected, smaller models (less layers/nodes per layer) will have lower train and test times in spite of outputting less reliable AoA estimations, but all will output the inferred AoA, given a single phase difference profile, in a very small fraction of a second ($\approx 1 \times 10^{-5}$ s).

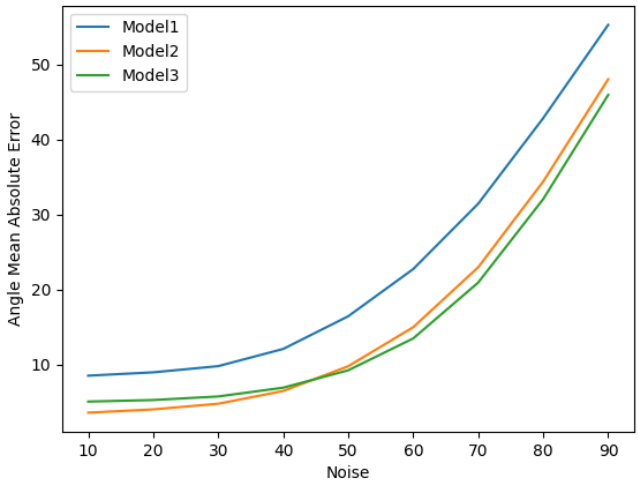

Figure 12: Mean Absolute Error (Real values)

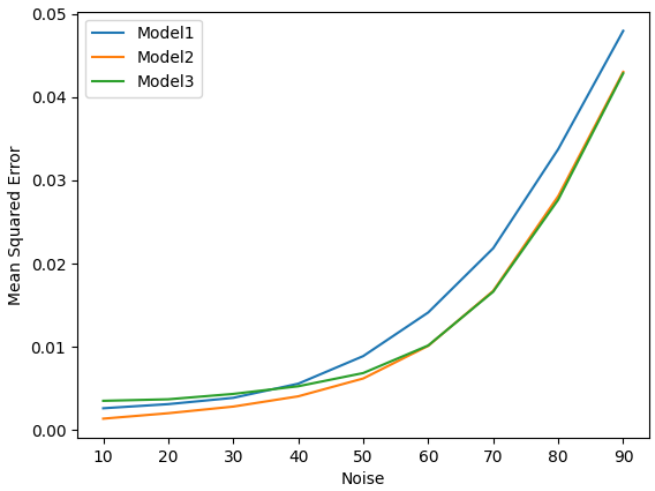

Figure 13: Mean Squared Error (Normalized values)

Model	Training time (20 epochs)	Testing time (all datasets)	Average testing time (per 72k datapoints)	AoA inference time
Model 1	1277.09 s	44.521 s	4.94 s	6.87×10^{-5} s
Model 2	1667.85 s	51.52 s	5.72 s	7.95×10^{-5} s
Model 3	2374.96 s	58.72 s	6.52 s	9.06×10^{-5} s

Table 1: Models temporal metrics

5 Conclusion

As the obtained results have shown, there is a trade-off between AoA estimation accuracy and training time, which won't be a burden since pre-trained models could then be deployed and applied to any environment. Additionally, in sufficiently controlled scenarios where computational speed is required with high levels of accuracy being disposable, the application of models 2 and 3 could be considered as estimations are reliable up to a certain noise level. In order to achieve better adapted models regarding their environment, the application of these in a reinforcement learning fashion (instead of supervised learning), where the model learns, via the environment it was installed on, as it performs, is also an area of further research. Also, noisy environments remain the top reason for unreliable inferences of, essentially, all AoA inferring methods, so future research on the topic is crucial towards achieving stable and clean indoor atmospheres.

References

[1] Nuno Paulino, Luís M. Pessoa, André Branquinho, and Edgar Gonçalves. Design and experimental evaluation of a bluetooth 5.1 antenna array for angle-of-arrival estimation, 2022.

[2] A.H. El Zooghby, C.G. Christodoulou, and M. Georgiopoulos. Performance of radial-basis function networks for direction of arrival estimation with antenna arrays. *IEEE Transactions on Antennas and Propagation*, 45(11):1611–1617, 1997.

[3] M. Pastorino and A. Randazzo. A smart antenna system for direction of arrival estimation based on a support vector regression. *IEEE Transactions on Antennas and Propagation*, 53(7):2161–2168, 2005.

[4] Aftab Khan, Stephen Wang, and Ziming Zhu. Angle-of-arrival estimation using an adaptive machine learning framework. *IEEE Communications Letters*, 23(2):294–297, 2019.

[5] Hongji Huang, Jie Yang, Hao Huang, Yiwei Song, and Guan Gui. Deep learning for super-resolution channel estimation and doa estimation based massive mimo system. *IEEE Transactions on Vehicular Technology*, 67(9):8549–8560, 2018.