A Direction of Arrival Machine Learning Approach for Beamforming in 6G

Anabel Reyes Carballeira
National Institute of
Telecommunications, INATEL
Santa Rita do Sapucai, MG, Brazil
email: anabel.carballeira@mtel.inatel.br

Abel Rodriguez Medel
National Institute of
Telecommunications, INATEL
Santa Rita do Sapucai, MG, Brazil
email: abel.medel@mtel.inatel.br

Jose M. Camara Brito
National Institute of
Telecommunications, INATEL
Santa Rita do Sapucai, MG, Brazil
email: brito@inatel.br

***Abstract*—Beamforming (BF) appropriately weights the amplitude and phase of individual antenna signals to create narrowly focused radiation. This makes it possible to provide better coverage in an indoor environment and at the edge of a cell. To make the best use of this technology, it is important to know the location of the device to direct the antenna beam of the radio Base Station (BS). Consequently, the Direction of Arrival (DOA) method is becoming very crucial and essential in this time. This paper proposes a Machine Learning (ML) based approach for DOA by evaluating three models: *Support Vector Classification* (SVC), *Decision Tree* (DT) and *Bagging Classifier* (BC). These models are trained using a public database built from drone's radio frequency signals. The proposed model significantly outperforms the techniques presented in previous work.**

***Index Terms*—Beamforming; Direction of Arrival; Machine Learning.**

I. INTRODUCTION

Up to the Fifth Generation Mobile Network (5G), the development of the cellular systems focused on communication aspects, while other services, had low priority. Diverging from current networks, future communication systems will become pervasive across multiple industry verticals by enabling a plethora of services that require location, such as assets tracking, context-aware marketing, transportation and logistics systems, Cross Reality (XR) experiences, and health care.

For location services, the Direction of Arrival (DOA) method estimates the direction angle of a source transmitting a signal to a receiver. DOA is highly applicable in wireless communications, astronomical observations, radar, and sonar [1]. In addition, with the beginning of the studies of the 6G network, the DOA methods assumes a new importance.

One of the key technologies in 6G is expected to be Beamforming (BF), specifically Holographic Beamforming (HBF) [2]–[5]. BF is a technique that focuses a wireless signal towards a specific receiving device, rather than having the signal spread in all directions from a broadcast antenna, as it usually would. Therefore, it is important to know the location of mobile devices and Internet of Things (IoT) terminals to direct the antenna beam of the radio BS. The resulting connection is faster and more reliable than it would be without BF.

The location technique for BF in 6G should be autonomous, reconfigurable, adaptive, and fast responsive. It is impossible to manually adjust the BF direction due to the extensive enhancement of the capacity of communication networks. ML has been considered part of the most important technologies in 6G [2], [6]–[10] due to the high capacity of communication networks and the massiveness of IoT devices.

ML is presented as a promising technology to be used for DOA. ML-based methods are data-driven and it can be more robust than other methods because they adapt better to array geometry imperfections and sensor imperfections. They also do not depend of the array geometry shape [1]. In addition, ML offers low-cost implementation and simplicity.

In [11], the authors propose a new DOA method based on an ML model to estimate the azimuth angle of a signal. The system employs only four antennas to find the direction of eight possible signal provenance angles. With this system was obtained a dataset named as *Dround_Data_New*, which contains well-known signals transmission powers for the eight angles. The authors trained and validated the dataset with a Deep Neural Network (DNN) model. The main goal of this work is to propose another ML model in order to increase the accuracy during prediction and decrease the training time over the same dataset *Dround_Data_New*.

The article is organized as follows. Section II is an overview of the related work. Section III extends the system architecture, the training dataset, the data modification for classification, and the proposed ML model. Section IV compares and discusses the results of the proposed DOA model with the approach presented in [11]. Finally, Section V concludes the paper.

To promote reproducible research, the code to generate the results of this paper is available here: [12].

II. RELATED WORK

Inherently multi-channel techniques, Multiple Signal Classification (MUSIC) [13], and Estimation of Signal Parameters Via Rotational Invariance Techniques (ESPRIT) [14], are considered to be the most popular DOA methods. However, there have been many problems associated with their application due to the base-band data from all antenna elements should be extracted simultaneously. A data correlation matrix needs to be formulated. Therefore, calibrating the values of each Radio Frequency (RF) channel becomes necessary to have a

ISBN: 978-1-61208-878-5

coherent multi-channel receiver [11]. A technique that does not require phase calibration is proposed in [15] as a Sparse Signal Representation (SSR) technique. However, this scheme cannot offer adequate accuracy.

In order to avoid the phase synchronization mechanism and antenna calibration mechanism, the authors in [11], [16], [17] used a system consisting of a single channel receiver, an M independent antennas, and a rotating switch similar to the one depicted in Figure 1. The authors in [16], examined the potential in using the different radiation patterns of a Switched Parasitic Antenna (SPA) for DOA. The DOA performance of the SPA is examined by calculating a lower bound on DOA accuracy, which is called the Cramer-Rao lower Bound (CRB). Then, all DOA estimation schemes derived from a general antenna array can also be applied to a parasitic antenna by inserting a new steering matrix. The authors in [17], employ a receiver, which is connected to one of M available sensors. Each sensor is enabled for reception every MT, where T is the time a sensor is enabled. The single-channel receiver has an RF switch that selects one sensor at a time. For the direction estimate, the posterior distribution of DOA is derived. Then, the Bayesian maximum posterior probability is applied to select the direction angle. The results of that work show high accuracy for the direction estimate of signals arriving from 10° and 40° angles simultaneously. However, the convergence time for the estimate is not clear because it is presented as a dimensionless quantity.

In [18], a comprehensive study is conducted to optimize the channel estimation and DOA estimation fields of massive Multiple Input Multiple Output (MIMO) based on the deep learning technique. Simulation results corroborate that the proposed scheme can achieve better performance in terms of DOA estimation compared with conventional methods. However, the system is complex and expensive because it employs 128 antennas. The authors in [19], proposed a cascaded neural network, which can be implemented to estimate the DOA of two closely spaced sources. The cascaded network consists of two parts: a SNR classification network and a DOA estimation network. The latter network contains two estimation subnetworks, which are applied to high and low SNRs to train the noisy data. The results of that work show better performance than other DOA techniques by detecting two signals coming from two directions separated by angles less than 5°. However, the performance is worse than the other techniques for sources with greater separation.

The authors in [11] used a system consisting of a single channel receiver, four antennas, and a rotating switch like the one depicted in Figure 1. There, the receiver sequentially activates the *n*-th antenna element at a time using the rotating switch, and measures the corresponding received power (P_n) where $n \in \{1, ..., N\}$. Then, the value of the normalized power measured in the *n*-th antenna will be given as:

$$x_n = \frac{P_n}{\sum_{i=1}^{N} P_i} \tag{1}$$

Next, the obtained normalized power values are fed to a Sparse Denoising Autoencoder (SDAE)-based DNN to find the azimuth angle of the transmitted signals. The SDAE makes compression and later a reconstruction of the power values to obtain a version of the power values with only essential information. Then, it starts the DNN training phase. DNN comprises three fully connected hidden layers and a softmax layer [20] for the classification task. The model achieved an overall accuracy of 96.25%. However, this paper proposes another ML method using the same dataset than in [11] to show improvements in both the overall accuracy and the elapsed training time for DOA.

III. System model and dataset

A. System overview

The system model consists of a single-channel receiver, and N directional antennas arranged circularly, see Figure 1. The antenna array is connected to the receiver using a non-reflective Single-Pole-N-Throw (SPNT) RF switch, which sequentially activates each antenna, and measures the received power values. The power measurements (P_1, P_2, P_3, ..., P_N) corresponding to each switching cycle are fed to the proposed ML model. Then, it is performed the DOA by exploiting the sparsity property of the incoming signal, and the gain variation property of the directional antenna array.

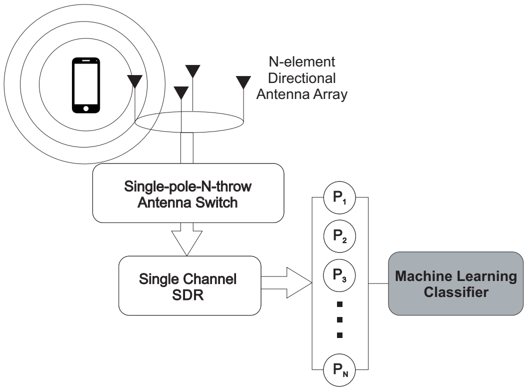


Fig. 1. The System Model.

B. Dround_Data_New dataset review

The *Dround_Data_New* dataset was collected by Software Defined Radio (SDR) (USRP B210), and a four-element sector antenna. The authors in [11], use only a single RF receiving channel for the SDR. Thus, the SDR is connected to the antenna using a non-reflective Single-Pole-4-Throw (SP4T) RF switch. The DJI Phantom 3 drone is the device that has been involved in this dataset. It is considered as the target device throughout the experiment. The drone downlink channel occupies the bandwidth from 2.401 GHz to 2.481 GHz. Each drone transmits a 10 MHz bandwidth Orthogonal Frequency Division Multiplexing (OFDM) signal. The OFDM signal transmitted by the drones is the main source to perform the DOA task. Figure 3a in [11] represents the environment that they used for the training data collection. The scenario represents a large ground with an open area. Also, there was a negligible RF intereference on the 2.0401 GHz - 2.481 GHz range due to the experiment occurring in a remote area.

To simplify the experiment, the authors in [11] virtually divided the area into eight octants (see Figure 3b in [11]). Therefore, *Dround_Data_New* is made up of octants. Each octant is considered as one direction (azimuth angle). For example, the first octant is considered as the angle's range 0°-45°, while the second octant is considered as the angle's range 45°-90°, and so on. Therefore, when the drone is flying, its direction is indicated by its corresponding octant. The dataset just has one type of data labeling for the classification of the azimuth angles. The *Dround_Data_New* dataset consists of 120 samples per angle's range. It is important to clarify that in the future more tests can be done and more data can be obtained to achieve a more precise direction. For example, instead of splitting the area into octants it can be splitted into sixteen or sixty-four parts.

C. Dataset modifications

It is common practice when performing a supervised ML experiment to hold out part of the available data as a validation set. The validation dataset is not used for training, instead, it is used to validate the trained model by predicting the labels of those unseen data. In [11], the dataset is divided into data to train and data to validate. The *Dround_Data_New* consists of 110 samples per angle for training and 10 samples for validation (representing approximately 8.33% of the total data). It is not a convincing validation dataset because it is small compared with the entire dataset due to the Pareto principle [21]. Besides, the data to train and validate were divided into different files in a fixed way, which makes it difficult to do different tests with different sizes of data for training and validation. Therefore, this data was joined in the same file and, shuffled, and splitted by using the *train_test_split* [22] method from the *Scikit-Learn (sklearn)* library to readjust dynamically the training and validation dataset sizes. In this work, the *train_test_split* method take as input: an array holding the samples, an array holding the class labels for the samples, a parameter called *test_size* which represents the proportion of the dataset to include in the validation dataset, and the parameter called *random_state* which controls the shuffling applied to the data before applying the split.

D. Classification model

In this subsection, a brief review of the ML model and the ensemble methods used in this work is given.

1) Support Vector Classification *(SVC):* SVC tries to find the best hyperplane to separate different classes by maximizing the distance between sample points and the hyperplane. The SVC model takes as input the following parameters:

- *kernel*: Selects the type of hyperplane used to separate the data. It must be one of *linear, poly, rbf, sigmoid, precomputed* or a callable.
- *C*: Is the penalty parameter of the error term. It controls the trade off between smooth decision boundary and classifying the training points correctly.
- *gamma*: Kernel coefficient for *rbf, poly* and *sigmoid.*

2) Decision Tree *(DT):* DT models are one of the simplest and most successful forms of ML models [23]. The goal of DT is to create a model that predicts the value of a target variable by learning simple rules inferred from the data features. The DT models build a tree during training that is the one applied when making the prediction. The input and output values can be discrete or continues. The DT model takes as input the following parameter:

- *max_depth*: This indicates how deep the tree can be.

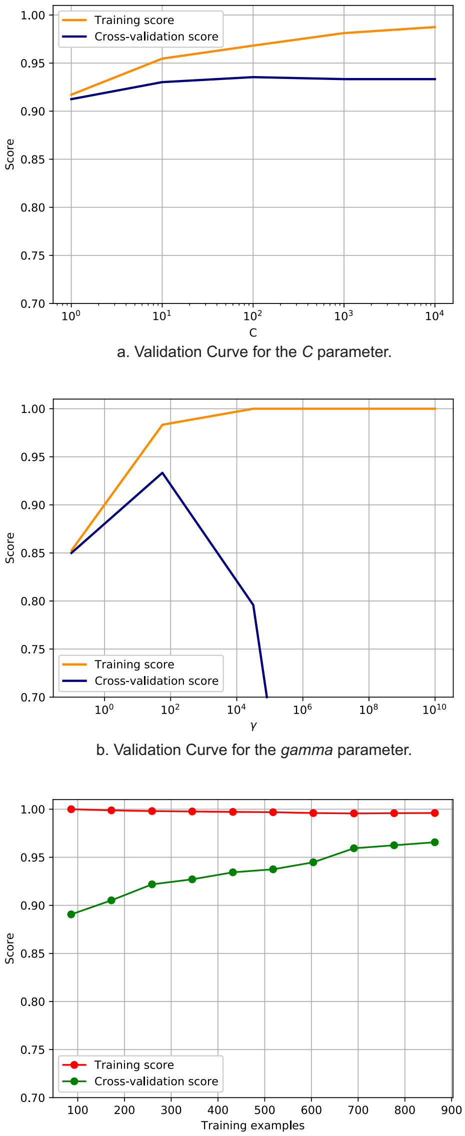
3) Bagging Classifier *(BC):* It is an ensemble meta-estimator that fits base classifiers each on random subsets of the original dataset and then aggregates their individual predictions (either by voting or by averaging) to form a final prediction [24]. This work uses the *Bagging Classifier* class from the *sklearn* library and it takes as input the following parameters:

- *base_estimator*: Applied to random subsets of the dataset. The base classifier used was DT [25].
- *n_estimators*: The number of base estimators (in this case, the number of DTs in the ensemble.
- *max_samples*: The number of samples to extract from the training data to train each base estimator.
- *bootstrap*: Defines whether samples are drawn with replacement. If *False*, sampling without replacement is performed.
- *n_jobs*: Tells Scikit-Learn the number of CPU cores to use for training and prediction. *n_jobs* is *None* by default, which means unset; it will generally be interpreted as *n_jobs*=1, which means that only one core will be used by Scikit-Learn. *n_jobs*=-1 tells Scikit-Learn library to use all available cores. For *n_jobs* below -1, number of cores + 1 + *n_jobs* are used. For example, with *n_jobs*=-2, all CPUs but one is used.
- *random_state*: Provided to control the random number generator used. The values of *random_state* can be: *None* (default), an integer, and a *numpy.random.RandomState* instance. *random_state=None* calls the function multiple times. It will reuse the same instance, and it will produce different results. If *random_state* is an integer, it is going to use a new random number generator seeded by the given integer. Using an integer will produce the same results across different calls. Popular integer random seeds are 0 and 42. The *numpy.random.RandomState* instance uses the provided random state, only affecting other users of that same random state instance.

IV. SIMULATION RESULTS

A. Tuning the parameters and the validation set

Figures 2 and 3 show the validation curves and learning curves for SVC and DT respectively. Figure 2a shows the behavior of the training score and validation score against the



a. Validation Curve for the *C* parameter.

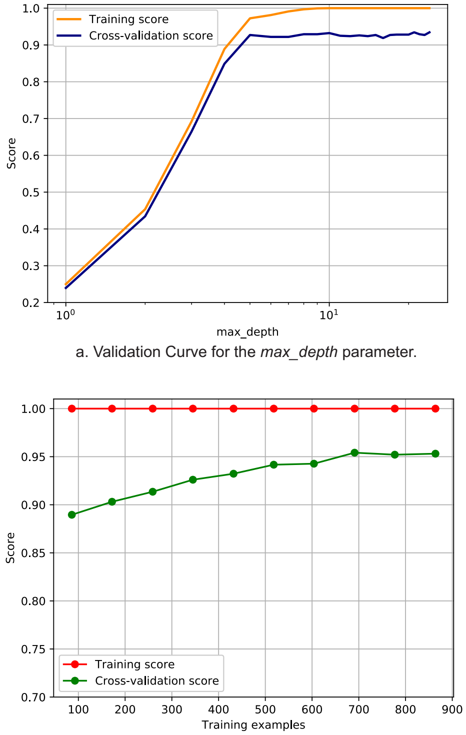
b. Validation Curve for the *gamma* parameter.

c. Learning Curve.

Fig. 2. Validations curves and Learning curves of SVC.

values of the parameter *C*. It is observed that the training score is always increasing and the validation score decreases slightly for a value of C equal to 100. Therefore, the value of C will be 100 to avoid overfitting, which means that the noise or random fluctuations in the training data is picked up and learned as concepts by the model. Figure 2b shows the training scores and validation scores of an SVC for different values of the kernel parameter *gamma*. Until *gamma* equal to approximately 100, it can be seen that both the training score and the validation score are increasing. After *gamma* equal to 100 the training score keeps increasing but the validation score decreases; therefore, the classifier overfit. The problem is that these concepts do not apply to new data and negatively impact the models ability to generalize.

Figure 3a shows that for low values of *max_depth* the DT model is underfitting, which means that it can neither model the training data nor generalize to new data. With the previous analysis and with the use of the search grid provided by *GridSearchCV* [26]—which exhaustively considers all parameter combinations to optimize a model— the best parameters for the SVC and DT models are summarized in Table I.



a. Validation Curve for the *max_depth* parameter.

b. Learning Curve.

Fig. 3. Validations curves and Learning curves of DT.

TABLE I
INPUT PARAMETER VALUES OF THE SCV AND DT MODELS.

Model	Parameters	Value
SVC	*kernel* *C* *gamma*	*rbf* 100 100
DT	*max_depth*	16

Figure 2c shows that the validation score is maximum after approximately 700 training samples and then remains almost constant, and the training score is still around the maximum. Therefore, 700 samples were destined for training, which represents 72.9% of the total samples (the total samples is 960). As a result of the total data, 70% was allocated for training and 30% to validate the SVC model. The same is observed in Figure 3b for the DT model. Therefore, 70% of the samples will also be used for training the DT model.

As BC is designed to reduce the possibility of overfitting complex models there is no need to analyze their validation curves. Table II shows the values given to the BC input parameters. The values of *n_estimators* and *max_samples* have been decided with the use of the search grid provided by *GridSearchCV* [26]. Figure 4 shows the learning curve of the BC model with a behavior similar to that of SVC and DT. Therefore, 70% of the data from *Dround_Data_New* dataset was also used to train the BC model.

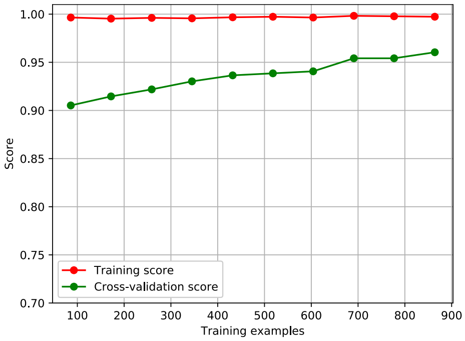


Fig. 4. Learning curves of BC.

TABLE II
INPUT PARAMETER VALUES OF THE *Bagging Classifier*.

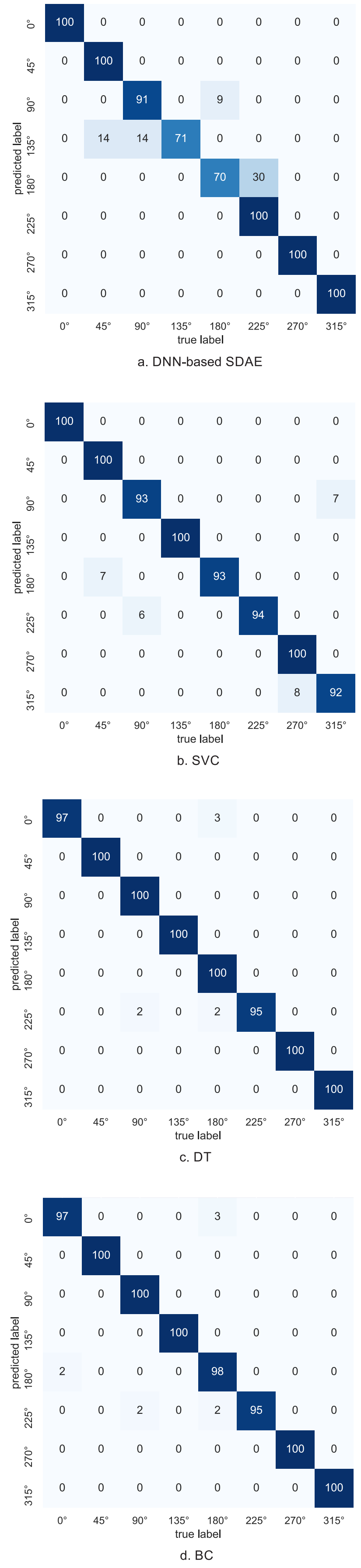
base_estimator	*Decision Trees*
n_estimators	300
max_samples	0.55
bootstrap	False
n_jobs	-1
random_state	42

B. Results

The performance of the SDAE-DNN model used in [11], SVC model, DT model and the BC model is shown in Figure 5 by means of confusion matrices and in Table III. It can be concluded that the DT and BC models present a better result than the SDAE based DNN and SVC models despite using less data to train. It can be seen that the BC and DT results are practically the same but with a longer time in BC (which makes sense as it is a more complex method). Therefore, the use of BC in this case does not make sense since DT provides the same results in a shorter time.

TABLE III
COMPARISON BETWEEN MODELS

Model	**Validation dataset size**	**Elapsed time**	**Accuracy**
SDAE-DNN	8.33%	109.29s	96.25%
SVC	10%	0.02s	95.83%
DT	30%	0.0s	98.61%
BC	30%	5.23s	98.61%



	0°	45°	90°	135°	180°	225°	270°	315°
0°	100	0	0	0	0	0	0	0
45°	0	100	0	0	0	0	0	0
90°	0	0	91	0	9	0	0	0
135°	0	14	14	71	0	0	0	0
180°	0	0	0	0	70	30	0	0
225°	0	0	0	0	0	100	0	0
270°	0	0	0	0	0	0	100	0
315°	0	0	0	0	0	0	0	100

a. DNN-based SDAE

	0°	45°	90°	135°	180°	225°	270°	315°
0°	100	0	0	0	0	0	0	0
45°	0	100	0	0	0	0	0	0
90°	0	0	93	0	0	0	0	7
135°	0	0	0	100	0	0	0	0
180°	0	7	0	0	93	0	0	0
225°	0	0	6	0	0	94	0	0
270°	0	0	0	0	0	0	100	0
315°	0	0	0	0	0	0	8	92

b. SVC

	0°	45°	90°	135°	180°	225°	270°	315°
0°	97	0	0	0	3	0	0	0
45°	0	100	0	0	0	0	0	0
90°	0	0	100	0	0	0	0	0
135°	0	0	0	100	0	0	0	0
180°	0	0	0	0	100	0	0	0
225°	0	0	2	0	2	95	0	0
270°	0	0	0	0	0	0	100	0
315°	0	0	0	0	0	0	0	100

c. DT

	0°	45°	90°	135°	180°	225°	270°	315°
0°	97	0	0	0	3	0	0	0
45°	0	100	0	0	0	0	0	0
90°	0	0	100	0	0	0	0	0
135°	0	0	0	100	0	0	0	0
180°	2	0	0	0	98	0	0	0
225°	0	0	2	0	2	95	0	0
270°	0	0	0	0	0	0	100	0
315°	0	0	0	0	0	0	0	100

d. BC

Fig. 5. Confusion matrix.

V. Conclusion

The necessity of having an intelligent system for DOA with reliable accuracy is becoming crucial for Next Generation Mobile Networks techniques like BF in 6G to improve the communication. These systems can integrate well-trained ML models to improve the robustness in performance. In this work, different ML models were trained using a public dataset. The best results were obtained for the DT model, overcoming another model proposed in the literature.

In future works, we will include not only the azimuth angle for DOA but also the elevation angle using the analyzed ML models. It will be considered an angle resolution of 5° instead of 45° to reduce the antenna beam and increase its directivity. The performance of the ML models will be evaluated by simulation results.

Acknowledgment

This work was partially supported by RNP, with resources from MCTIC, Grant No. No 01245.010604/2020-14, under the 6G Mobile Communications Systems project of the Radiocommunication Reference Center (Centro de Referência em Radiocomunicações - CRR) of the National Institute of Telecommunications (Instituto Nacional de Telecomunicações - Inatel), Brazil.

References

[1] Z.-M. Liu, C. Zhang, and S. Y. Philip, "Direction-of-arrival estimation based on deep neural networks with robustness to array imperfections," *IEEE Transactions on Antennas and Propagation*, vol. 66, no. 12, pp. 7315–7327, 2018.

[2] Z. Zhang, Y. Xiao, Z. Ma, M. Xiao, Z. Ding, X. Lei, G. K. Karagiannidis, and P. Fan, "6g wireless networks: Vision, requirements, architecture, and key technologies," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 28–41, 2019.

[3] B. Zong, C. Fan, X. Wang, X. Duan, B. Wang, and J. Wang, "6g technologies: Key drivers, core requirements, system architectures, and enabling technologies," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 18–27, 2019.

[4] M. Z. Chowdhury, M. Shahjalal, S. Ahmed, and Y. M. Jang, "6g wireless communication systems: Applications, requirements, technologies, challenges, and research directions," *IEEE Open Journal of the Communications Society*, vol. 1, pp. 957–975, 2020.

[5] S. Elmeadawy and R. M. Shubair, "Enabling technologies for 6g future wireless communications: Opportunities and challenges," *arXiv preprint arXiv:2002.06068*, 2020.

[6] K. David and H. Berndt, "6g vision and requirements: Is there any need for beyond 5g?," *IEEE Vehicular Technology Magazine*, vol. 13, no. 3, pp. 72–80, 2018.

[7] S. J. Nawaz, S. K. Sharma, S. Wyne, M. N. Patwary, and M. Asaduzzaman, "Quantum machine learning for 6g communication networks: State-of-the-art and vision for the future," *IEEE Access*, vol. 7, pp. 46317–46350, 2019.

[8] K. B. Letaief, W. Chen, Y. Shi, J. Zhang, and Y.-J. A. Zhang, "The roadmap to 6g: Ai empowered wireless networks," *IEEE Communications Magazine*, vol. 57, no. 8, pp. 84–90, 2019.

[9] E. C. Strinati, S. Barbarossa, J. L. Gonzalez-Jimenez, D. Ktenas, N. Cassiau, L. Maret, and C. Dehos, "6g: The next frontier: From holographic messaging to artificial intelligence using subterahertz and visible light communication," *IEEE Vehicular Technology Magazine*, vol. 14, no. 3, pp. 42–50, 2019.

[10] F. Tariq, M. R. Khandaker, K.-K. Wong, M. A. Imran, M. Bennis, and M. Debbah, "A speculative study on 6g," *IEEE Wireless Communications*, vol. 27, no. 4, pp. 118–125, 2020.

[11] S. Abeywickrama, L. Jayasinghe, H. Fu, S. Nissanka, and C. Yuen, "Rf-based direction finding of uavs using dnn," in *2018 IEEE International Conference on Communication Systems (ICCS)*, pp. 157–161, IEEE, 2018.

[12] (accessed July 5, 2021). https://github.com/anakings/DOA-ML/tree/master/my_solution.

[13] R. Schmidt, "Multiple emitter location and signal parameter estimation," *IEEE transactions on antennas and propagation*, vol. 34, no. 3, pp. 276–280, 1986.

[14] R. Roy and T. Kailath, "Esprit-estimation of signal parameters via rotational invariance techniques," *IEEE Transactions on acoustics, speech, and signal processing*, vol. 37, no. 7, pp. 984–995, 1989.

[15] H. Fu, S. Abeywickrama, C. Yuen, and M. Zhang, "A robust phase-ambiguity-immune doa estimation scheme for antenna array," *IEEE Transactions on Vehicular Technology*, vol. 68, no. 7, pp. 6686–6696, 2019.

[16] T. Svantesson and M. Wennstrom, "High-resolution direction finding using a switched parasitic antenna," in *Proceedings of the 11th IEEE Signal Processing Workshop on Statistical Signal Processing (Cat. No. 01TH8563)*, pp. 508–511, IEEE, 2001.

[17] Q. Gao, Y. Zhao, and D. Hu, "Direction-of-arrival estimation for wideband array signals with a single channel receiver," in *2014 IEEE Radar Conference*, pp. 0809–0812, IEEE, 2014.

[18] H. Huang, J. Yang, H. Huang, Y. Song, and G. Gui, "Deep learning for super-resolution channel estimation and doa estimation based massive mimo system," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 9, pp. 8549–8560, 2018.

[19] Y. Guo, Z. Zhang, Y. Huang, and P. Zhang, "Doa estimation method based on cascaded neural network for two closely spaced sources," *IEEE Signal Processing Letters*, vol. 27, pp. 570–574, 2020.

[20] B. Yuan, "Efficient hardware architecture of softmax layer in deep neural network," in *2016 29th IEEE International System-on-Chip Conference (SOCC)*, pp. 323–326, IEEE, 2016.

[21] R. Sanders, "The pareto principle: its use and abuse," *Journal of Services Marketing*, 1987.

[22] (accessed July 5, 2021). https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.train_test_split.html.

[23] G. Brewka, "Artificial intelligence—a modern approach by stuart russell and peter norvig, prentice hall. series in artificial intelligence, englewood cliffs, nj.," *The Knowledge Engineering Review*, vol. 11, no. 1, pp. 78–79, 1996.

[24] (accessed July 5, 2021). https://scikit-learn.org/stable/modules/generated/sklearn.ensemble.BaggingClassifier.html#:~:text=A%20Bagging%20classifier%20is%20an,to%20form%20a%20final%20prediction.

[25] (accessed July 5, 2021). https://scikit-learn.org/stable/modules/tree.html.

[26] (accessed July 5, 2021). https://scikit-learn.org/stable/modules/generated/sklearn.model_selection.GridSearchCV.html.