# Designing Smart Antennas Using Machine Learning Algorithms

Barsa Samantaray[1], Kunal Kumar Das[1], and Jibendu Sekhar Roy[2]

[1]*SOA University, Bhubaneswar, India,*
[2]*KIIT University, Bhubaneswar, Odisha, India*

https://doi.org/10.26636/jtit.2023.4.1329

**Abstract — Smart antenna technologies improve spectral efficiency, security, energy efficiency, and overall service quality in cellular networks by utilizing signal processing algorithms that provide radiation beams to users while producing nulls for interferers. In this paper, the performance of such ML solutions as the support vector machine (SVM) algorithm, the artificial neural network (ANN), the ensemble algorithm (EA), and the decision tree (DT) algorithm used for forming the beam of smart antennas are compared. A smart antenna array made up of 10 half-wave dipoles is considered. The ANN method is better than the remaining approaches when it comes to achieving beam and null directions, whereas EA offers better performance in terms of reducing the side lobe level (SLL). The maximum SLL is achieved using EA for all the user directions. The performance of the ANN algorithm in terms of forming the beam of a smart antenna is also compared with that of the variable-step size adaptive algorithm.**

***Keywords — artificial neural network, decision tree, ensemble algorithm, machine learning, smart antenna, support vector machine***

## 1. Introduction and Related Work

A smart antenna system (SAS) is an adaptive aerial array that radiates a beam towards the user and a null towards the interferer in a cellular network, after estimating the direction of arrival (DOA) of the signal from a mobile device. The system uses a signal processing algorithm [1], [2] to create such an adaptive system. SAS may have the form of a switched beam antenna or an adaptive antenna solution [2], [3]. The switched beam system forms the beam only in pre-defined directions, whereas the adaptive beam system forms the beam in any desired direction. The adaptive SAS concept is depicted in Fig. 1. In this approach, the signal's DOA is estimated first, before forming a retro-directive main beam towards the desired users. The performance of the adaptive algorithm is the main factor in the performance of the entire SAS.

There are several adaptive signal processing algorithms [4], [5], each of them having its specific pros and cons. The most common algorithms used for SAS-based beamforming include the least mean square (LMS) algorithm [6], [7], the recursive least square (RLS) algorithm [8] and the sample matrix inverse (SMI) algorithm [9]. The variable step-size LMS algorithm is used for beamforming in [6], where lower SLL values are achieved. In [7], various configurations are used in a SAS, relying on the LMS algorithm for 5G and 6G

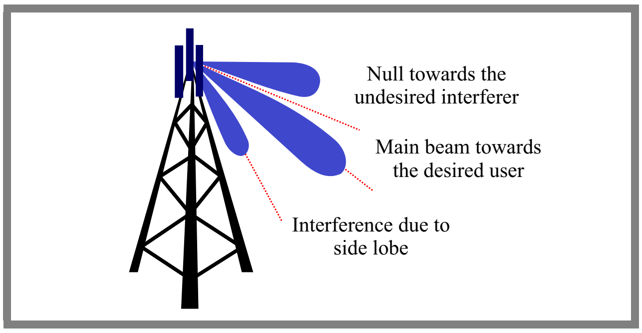


**Fig. 1.** Smart antenna concept.

energy harvesting applications. The variable step-size LMS (VS-LMS) algorithm is used for beamforming in [6], where a lower SLL value is achieved. In [7], various dipole configurations are used in a SAS solution with the LMS algorithm for 5G and 6G energy harvesting applications. The tracking properties of the RLS algorithm are used in [8] for an adaptive antenna operating in a flat Rayleigh fading environment, to solve the interference cancellation problem. To avoid the drawbacks of traditional algorithms, the adaptive diagonal loading SMI algorithm is used in smart antennas to enhance their performance in varying signal-to-noise (SNR) conditions [9].

Many other algorithms are applied for beamforming in SAS, e.g. those presented in [10], [11]. The minimum variance distortion less response (MVDR) approach to beamforming, which is capable of estimating the weight vectors for adaptive beam steering, is used in [10] to improve cellular network capacity. The constant modulus algorithm (CMA) is used for blind adaptive beam formation and for SLL reduction in paper [11].

The applications of ML in smart antenna beamforming processes are relatively new and examples of using such a technique are presented, for instance, in [12], [13]. In addition, the majority of papers deal with isotropic antennas. The use of ANN for designing a SAS is considered in paper [14] and serves as a remedy for the complexity of the antenna array's design stemming from its nonlinear nature. Good performance of ANN in terms of improving directivity and reducing SLL of smart solutions with circular and concentric circular arrays of isotropic antennas is reported in [15]. An overview of different ML types and the importance of applying this method while modelling antennas and antenna arrays are briefly pre-

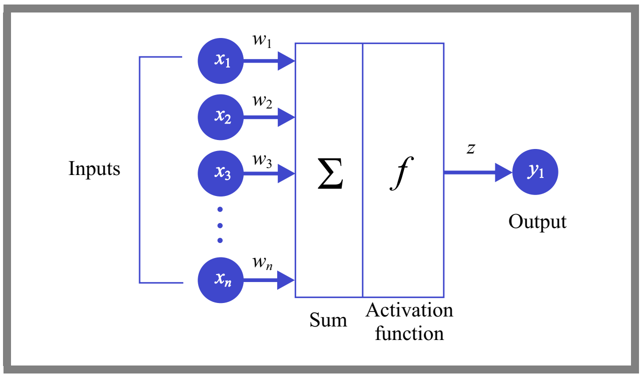


**Fig. 2.** Simulation model for ANN.

sented in [16]. An intelligent synthesis method based on SVM for antennas and smart antennas is presented in [17], where an antenna classification score of over 99% and a parameter prediction with a mean absolute percentage error of less than 6% were achieved.

The ML method using an ensemble model which combines two or more models for better output is used for the prediction of the bandwidth of metamaterial antennas in [18], where the model's data are processed using DT and SVM algorithms.

According to related work, signal processing algorithms seem to be used for the purpose of designing SAS in the majority of cases. The use of machine learning allows to avoid the deployment of digital signal processors (DSP). The aim of this paper is to investigate the performance of ML algorithms in the design of SAS of a dipole array. For beam formation of smart antennas, four types of ML algorithms are used: SVM, ANN, EA, and DT, and their performance is compared. As far as the achievement of the desired beam and null directions is concerned, ANN offers better performance, whereas the EA method is better for reducing SLL.

## 2. Machine Learning Algorithms

Machine learning (ML) techniques have been developing rapidly in recent years and are increasingly used for analyzing and computing data in such fields as mobile communications, healthcare, the Internet of Things (IoT), social media, industrial applications, and so on. ML is the key technology used for intelligent classification and analysis of data in real-world applications. The ML domain may be divided classified broadly into four types of learning: supervised, unsupervised, semi-supervised, and reinforcement learning [19], [20]. In supervised learning, the operator of the ML method supplies a dataset with known inputs and outputs, and the algorithm is used to find values that are the closest to those inputs and outputs. In unsupervised learning, the ML method does not rely on an answer key or a human operator. Here, the machine estimates the correlations by analyzing the pattern of the available data. Semi-supervised learning is the intermediate state between supervised and unsupervised learning algorithms, where both labeled and unlabeled datasets are used during the training phase. In reinforcement learning, a set of actions, parameters, and end values is provided by the ML algorithm. Then the ML method explores the best optimal result using the trial-and-error approach.

The supervised SVM algorithm is used for both classification and regression [21]. In the decision function, known as support vectors, a subset of training points is used to make it memory efficient. In SVM, the regression models are based on finding a function $f(x)$ that satisfies the coefficients of an equation and has a minimum difference between the actual and predicted responses for the training data under consideration. SVM is a linear algorithm and may be even classified as a linear regression approach. An SVM classifier creates a line (a plane or a hyper-plane, depending upon the dimensions of the data set) in an $N$-dimensional space to classify data points into two separate classes. If the data set contains more noise, SVM does not perform well. In this work, a linear SVM with sequential minimal optimization (SMO) [22] having a linear kernel polynomial of order 3, is used.

ANN is an algorithm that is derived from the biological neural network of the human brain [23], [24] and is a sub-category of deep learning. ANN is a fully connected multi-layer neural network having an input layer, multiple hidden layers, and an output layer. A simple model of an artificial neuron is shown in Fig. 2, where random weights are used and the weighted sum of inputs is passed through an activation function of a non-linear nature. The signal flows from the left to the right -– a move known as "forward pass". The output is compared with the training data, and then the error is calculated. Next, the network performs the "backward pass" from the right to the left and propagates the error to every individual node using the back propagation technique. Accordingly, weights are adjusted to reduce the error unless the required output is achieved.

In EA, decisions from several models are combined to improve overall performance [25], [26]. Here, the supervised learning process as well as the classification and regression tasks are performed by linear combination, and the outputs of the trained base learner are real-valued probability estimates of the class label given the input data. The combination of these base learners is expressed as an ensemble probability estimate, such as:

$$p(y|x) = \sum_{t=1}^{T} w_t p_t(y|x) \ , \tag{1}$$

where $p_t(x|y)$ is the probability of estimating class label $y$ for a given input $x$ by the trained base learner $t$ with weight $w_t$. For a uniformly distributed base learner, $w_t = 1/T$. Compared with the basic methods, the performance of ensemble learning methods for decision-making processes is very good. In this work, the least-square boost ensemble algorithm [27] is used.

The DT algorithm is a supervised ML method and is mostly used to solve classification problems, though it can be used for regression problems as well [28], [29]. DT relies on decision nodes and leaf nodes. Decision nodes are used to take decisions and have many branches, whereas leaf nodes are the outputs of those decisions and have no branches. The possible solutions to a problem or decision are represented graphical-

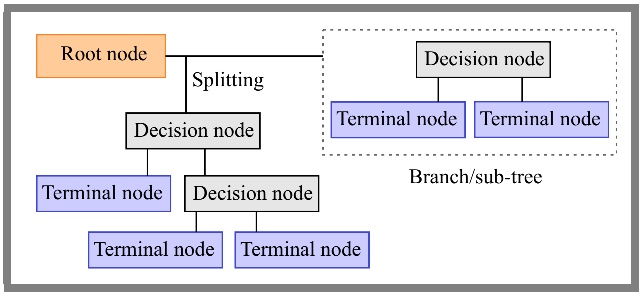


**Fig. 3.** DT algorithm model.

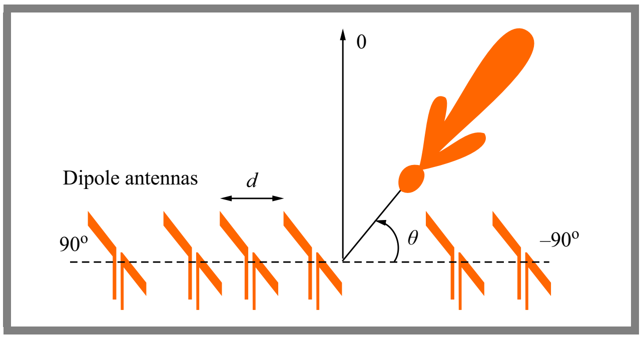


**Fig. 4.** Linear dipole antenna array.

ly for specific, given conditions. DT contains a number of terminologies. The root node is the entire population or sample, which is further divided into two or more homogeneous sets. In the splitting phase, the process is divided into two or multiple sub-nodes, and when the split node is divided further, it is called the decision node. Terminal or leaf nodes do not split any further. The split node is known as the parent node, and the sub-node is known as the child node. When the sub-nodes of a decision node are removed, the process is called pruning.

Basic DT architecture is shown in Fig. 3. The different instances are classified by DT by estimating the attributes of the nodes, starting from the root node all the way to the tree branch nodes corresponding to those attribute values. The splitting is based on Gini impurity and entropy criteria used for estimating the information gain. Entropy $H(x)$ is expressed for the probability $p(x)$ as [29]:

$$H(x) = -\sum_{i=1}^{n} p(x_i)\log_2 p(x_i) \ , \tag{2}$$

and Gini $(E)$ is defined as:

$$E = 1 - \sum_{i=1}^{c} p_i^2 \ . \tag{3}$$

In this work, deep regression [30] is used. In deep regression, the features of deep neural networks are integrated in the course of a regression analysis.

## 3. Beamforming of SAS Using ML

In this work, a smart antenna of made of linear, uniform, half-wave dipole antennas is considered (Fig. 4). The antennas are separated by uniform spacing of $d$. For a dipole of length of

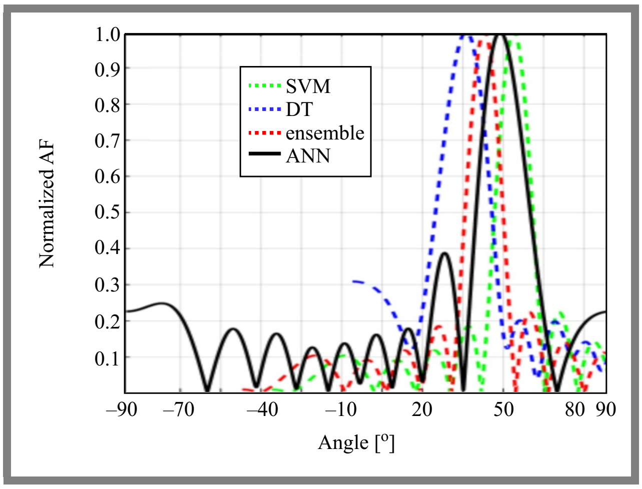


**Fig. 5.** Comparison of results for BD=47° and ND=35°.

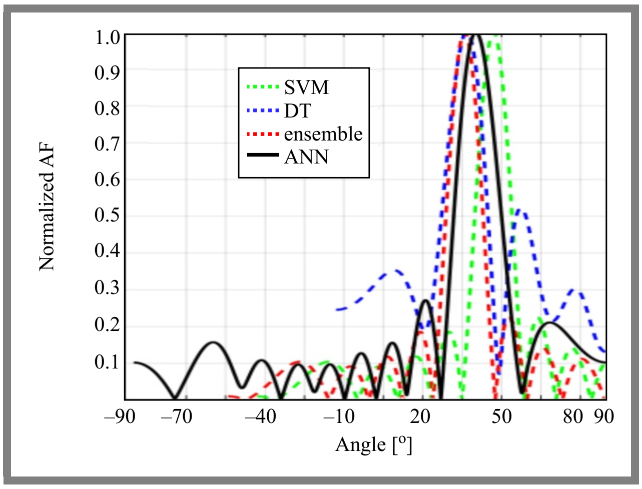


**Fig. 6.** Comparison of results for BD=40° and ND=27°.

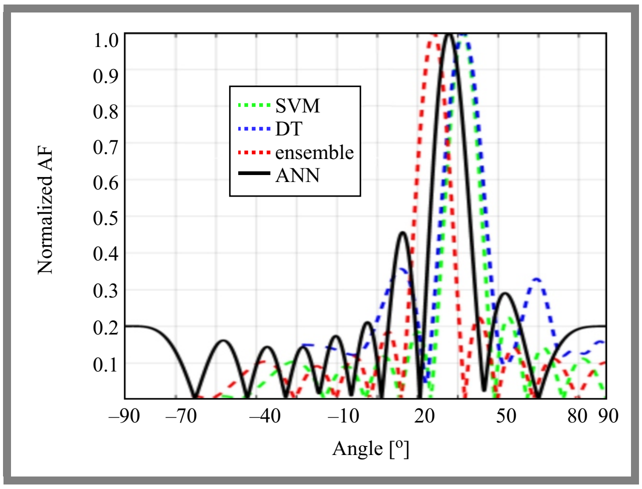


**Fig. 7.** Comparison of results for BD=30° and ND=21°.

$l$, the radiated electric field is [3]:

$$E(\theta) = j\eta \frac{I_0 e^{-j\beta r}}{2\pi r}\left[\frac{\cos(\frac{\beta l}{2}\cos\theta) - \cos\frac{\beta l}{2}}{\sin\theta}\right] , \tag{4}$$

where $\beta = 2\pi/\lambda$ is the propagation constant, $I_0$ is the current amplitude, $\eta = 120\ \pi\Omega$ is the free space impedance and $r$ is the distance between the observation point and the source.

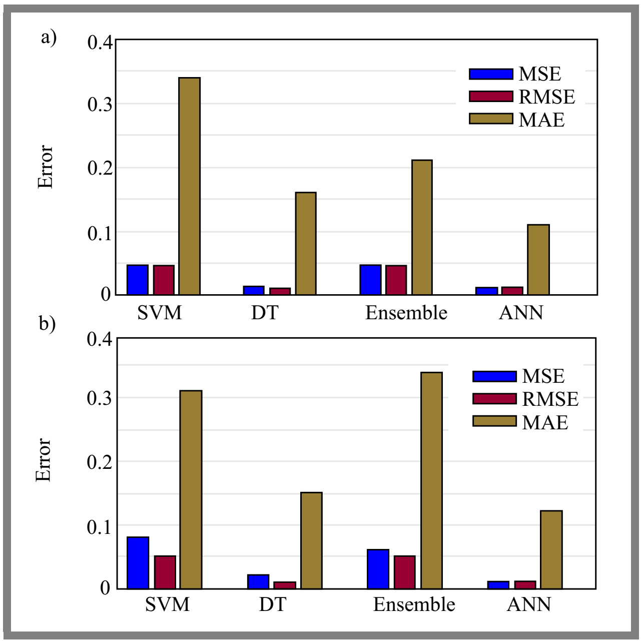


**Fig. 8.** Graphs showing errors achieved using ML methods for: a) BD=40°, ND=27° and b) BD=47°, ND=35°.

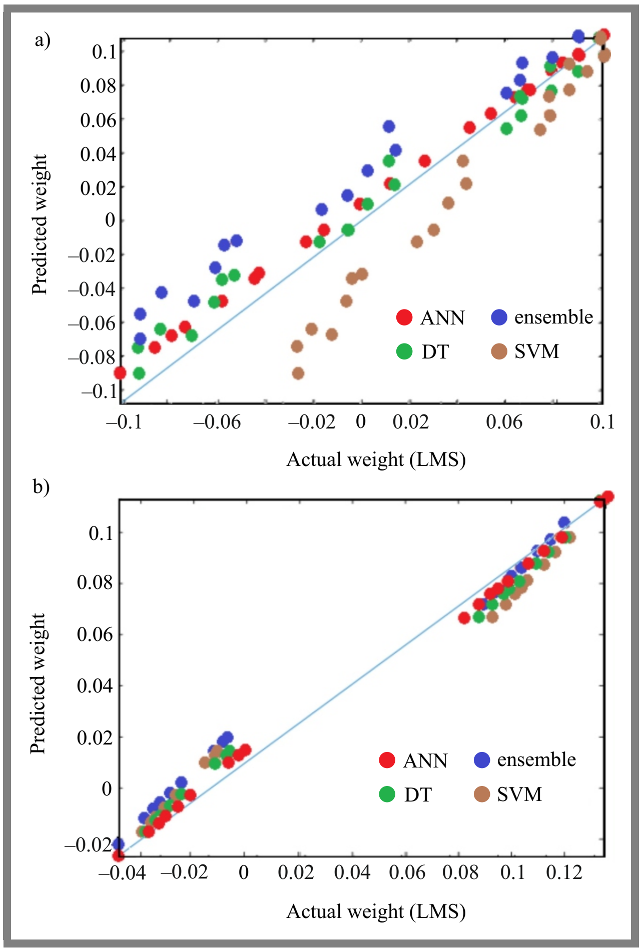


**Fig. 9.** Predicted weight vs. actual weight for: a) BD=40°, ND=27° and b) BD=47°, ND=35°.

For $N$ dipoles in the array, the total radiated field is:

$$E_{total} = E(\theta)AF(\theta) \ , \tag{5}$$

**Tab. 1.** Comparison of results obtained using different ML methods.

| Desired BD and ND | ML method | Obtained BD [°] | Obtained ND [°] | SLLmax [dB] |
|---|---|---|---|---|
| BD=47° ND=35° | SVM | 53.51 | 41.82 | –9.12 |
| | DT | 34.88 | 19.98 | –10.46 |
| | Ensemble | 44.21 | 33.01 | –13.42 |
| | ANN | 46.42 | 35.02 | –8.40 |
| BD=40° ND=27° | SVM | 47.21 | 34.91 | –12.40 |
| | DT | 34.84 | 20.00 | –5.68 |
| | Ensemble | 34.85 | 22.15 | –13.65 |
| | ANN | 39.50 | 26.44 | –11.32 |
| BD=30° ND=21° | SVM | 36.00 | 20.00 | –12.40 |
| | DT | 36.00 | 19.95 | –8.87 |
| | Ensemble | 25.68 | 15.22 | –13.55 |
| | ANN | 30.42 | 20.44 | –6.74 |

where $AF(\theta)$ is the array factor for isotropic elements given by:

$$AF(\theta) = \sum_{n=1}^{N} I_0 \, e^{j(n-1)(\frac{2\pi d}{\lambda}\cos\theta+\alpha)} \ , \tag{6}$$

where $\lambda$ is the wavelength, $\alpha$ is the progressive phase shift of the dipole array and $I_0$ is the feed current. Equations (5), (6) are the cost function for beamforming using ML methods.

In this work, a uniform linear array of 10 ($N = 10$) half-wavelength dipoles with inter-element spacing of $d = \lambda/2$ is considered. The frequency of operation is 1800 MHz ($f = 1800$ MHz), the desired beam (user) direction (BD) is $\theta_s$ and the desired null (interferer) direction (ND) is $\theta_I$.

For beamforming using the ML method, a data set with a matrix dimension of (2205×25) is created from different combinations of $f$, $d$, $\beta$, $\theta_s$ and $\theta_I$. The ANN network consists of 5 input variables, 2 hidden layers and 20 output variables. The first hidden layer has 20 neurons and the second hidden layer has 5 neurons. The ANN is run 9000 times with the learning rate initialization of 0.01 and a tolerance value of $10^{-5}$. After training and testing, 10 complex weights are obtained and these updated weights are used for smart antenna beam formation using the cost function, i.e. the array factor for the dipole array. For beamforming using SVM, a linear SVM model is created by using solver-sequential minimal optimization (SMO) with a linear kernel of the third order polynomial. In SVM, after training and testing, 10 complex weights are obtained and the process is run 9000 times. For DT implementation, a deep regression decision tree model [30] is created and other parameters, such as SVM, are estimated. For the ensemble algorithm, a boost ensemble model [27] is created and other parameters, such as SVM, are assumed.

Matlab is used to simulate the beamforming of smart antennas of dipole arrays with the use of ML methods. Normalized array factors (AF) for a smart antenna comprising a 10-element dipole array are compared in Figs. 5 – 7, for different beam directions (BD) and null directions (ND). The plots showing mean square error (MSE), root mean square error

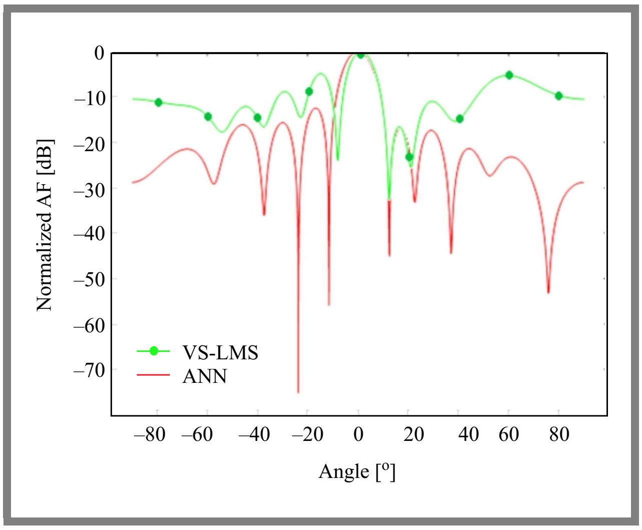


**Fig. 10.** Simulation results for ANN and VS-LMS (BD=0°, ND=12°).

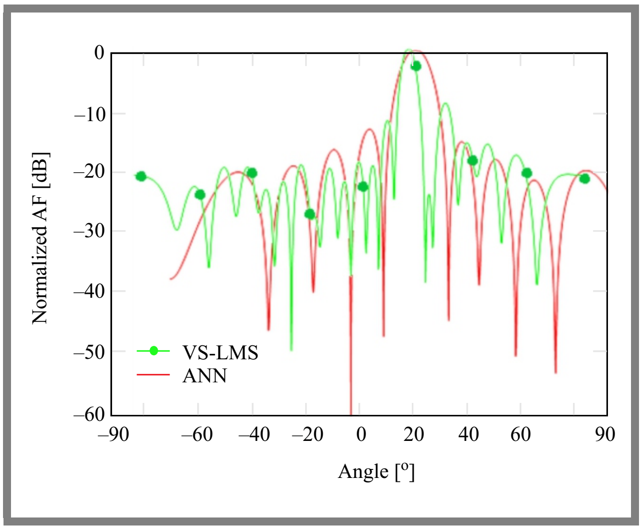


**Fig. 11.** Simulation results for ANN and VS-LMS (BD=20°, ND=32°).

(RMSE) and mean absolute error (MAE) are shown in Fig. 8. The predicted weight and actual weight for ANN, ensemble, DT, and SVM methods are shown in Fig. 9. In Fig. 9, the difference between actual and predicted weights are illustrated for ANN and other ML methods. The results for beamforming of a smart antenna are shown, for reference, in Tab. 1.

In the next step, parameters of the beamforming process relying on the ANN method is compared with the method based on the signal processing algorithm, i.e., variable step-size LMS (VS-LMS) algorithm [6], [31]. In VS-LMS, step size $\mu$ is a variable and is evaluated as [31]:

$$\mu_{n+1} = \begin{cases} \alpha\mu_n + \delta\varepsilon_n \ , & \text{if} \quad 0 < \mu_{n+1} < \mu_{max} \\ \mu_{max} \ , & \text{otherwise} \end{cases} \ , \tag{7}$$

where $\mu_{max}$ is the maximum value of the step-size parameter [6], [31]. In Eq. (7), $\alpha$ and $\delta$ are constant parameters and, in this work, $\alpha = 0.95$ and $\delta = 0.0003$. Factor $\varepsilon_n$ in Eq. (7) is related to the weight vectors [31].

The VS-LMS algorithm updates the weight according to the following relation:

$$w(n+1) = w(n) + \mu_{n+1} x(n) e^*(n) \ , \tag{8}$$

where $e^*(n)$ is the complex conjugate of the error $e(n)$ between the desired signal $d(n)$ and array output $y(n)$. The performance of ANN and VS-LMS for beamforming in a smart antenna is compared in Figs. 10–11. In both cases (Fig. 10 and Fig. 11), ANN shows better performance in terms of reducing SLL. The maximum SLL values in Fig. 10, using ANN and VS-LMS, are –13.3 dB and –4.7 dB, respectively. In Fig. 11, the maximum SLL values using ANN and VS-LMS, are –13.4 dB and –9 dB, respectively.

The results for maximum SLL (SLLmax) are compared with other papers [32]–[34] in Tab. 2. A recurrent neural network (RNN) from [32], is used for beamforming. The RNN is based on the gated recurrent umit, the Elman RNN, and the SVM with the covariance matrix taper from [33]–[35], respectively. No relevant papers on the application of DT and EA algorithms for beamforming in smart antennas were found in recent publications.

**Tab. 2.** Performance comparison: ANN vs. other papers.

| References | Beamforming method | Parameters | SLLmax [dB] |
|---|---|---|---|
| [32] | RNN | $N$=32, BD=0° | –7.5 |
| | RNN | $N$=16, BD=–10° | –8.5 |
| [33] | RNN based on the gated recurrent unit | $N$=16, BD=100° | –11.5 |
| [34] | Elman RNN | $N$=5, BD=30° | –11.5 |
| [35] | SVM with covariance matrix taper | $N$=10, BD=0° | –10.5 |
| This paper | ANN | $N$=10, BD=40° | –11.32 |
| | SVM | $N$=10, BD=30° | –12.40 |
| | DT | $N$=10, BD=47° | –10.46 |
| | EA | $N$=10, BD=40° | –13.65 |

Side lobe is one of the main reasons generating interference in cellular networks. The reduction in SLL presented in this paper is compared with other reported results.

# 4. Conclusions

For a smart antenna made of a dipole array designed using ML algorithms, ANN shows outperforms other methods in terms of achieving the desired null and beam directions, whereas the EA method is better suited for reducing SLL. SLL reduction is important for minimizing interference in a cellular network. The theoretical aspects and the simulation methods are easier for the ANN approach when compared with other ML algorithms. The simulation time for the ANN method is the shortest compared with other methods. For BD=47$^{o}$ and ND= 35$^{o}$, the simulation using SVM, DT, EA, and ANN methods took 5.724, 1.057, 1.174, and 1.022 minutes, respectively. ML algorithms may be successfully applied beamforming in smart antennas used in cellular networks, including massive multiple-input multiple-output (MIMO) systems operating in multiuser environments, where antenna arrays are used with a very narrow beam pointing towards the target.

## References


[1] C.A. Balanis, S. Bellofiore, J. Foutz, and A.S. Spanias, "Smart Antenna System for Mobile Communication Network", *IEEE Antenna and Propagation Magazine*, vol. 44, no. 4, pp. 106–114, 2002 (https://doi.org/10.1109/MAP.2002.1043158).

[2] T.K. Sarkar, M.C. Wicks, M. Salazar-Palma, and R.J. Bonneu, *Smart Antennas*, Wiley IEEE Press, 2003 (ISBN: 0471210102).

[3] C.A. Balanis, *Antenna Theory: Analysis and Design*, 3rd ed., Wiley, Hoboken, USA, 2005 (ISBN: 9780471667827).

[4] M. Abualhayja'a and M. Houssain, "Comparative Study of Adaptive Beamforming Algorithms for Smart Antenna Applications", *International Conference on Communications, Signal Processing, and their Applications (ICCSPA)*, Sharjah, UAE, 2021 (https://doi.org/10.1109/ICCSPA49915.2021.9385725).

[5] M. Atzemourt, A. Farchi, Y. Chihab, and Z. Hachkar, "Performance Evaluation of LMS and CM Algorithms for Beamforming", *Advances in Materials Science and Engineering*, vol. 2022, art. no. 7744625, 2022 (https://doi.org/10.1155/2022/7744625).

[6] B. Samantaray, K.K. Das, and J.S. Roy, "Performance of Smart Antenna in Cellular Network Using Variable Step Size Algorithms", *International Journal of Microwave and Optical Technology (IJMOT)*, vol. 15, no. 2, pp. 179–186, 2020 (https://www.ijmot.com/VOL-15-NO-2.aspx).

[7] K. Pirapaharan, N. Ajithkumar, K. Sarujan, X. Fernando, and P.R.P. Hoole, "Smart, Fast, and Low Memory Beam-steering Antenna Configurations for 5G and Future Wireless Systems", *Electronics*, vol. 11, no. 17, art no. 2658, 2022 (https://doi.org/10.3390/electronics11172658).

[8] B.C. Banister and J.R. Zeidler, "Tracking Performance of the RLS Algorithm Applied to an Antenna Array in a Realistic Fading Environment", *IEEE Transactions on Signal Processing*, vol. 50, no. 5, pp. 1037–1050, 2002 (https://doi.org/10.1109/78.995061).

[9] J. Gao, J. Zhen, Y. Lv, and B. Guo, "Beamforming Technique Based on Adaptive Diagonal Loading in Wireless Access Networks", *Ad Hoc Networks*, vol. 107, art. no. 102249, 2020 (https://doi.org/10.1016/j.adhoc.2020.102249).

[10] B. Boustani, A. Baghdad, A. Sahel, A. Badri, and A. Ballouk, "Adaptive Algorithms for Smart Antenna System", *2018 6th International Conference on Multimedia Computing and Systems (ICMCS)*, Rabat, Morocco, 2018 (https://doi.org/10.1109/ICMCS.2018.8525905).

[11] Veerendra, M. Bakhar, and R.M. Vani, "Robust Blind Beam Formers for Smart Antenna System Using Window Techniques", *Procedia Computer Science*, vol. 93, pp. 713–720, 2016 (https://doi.org/10.1016/j.procs.2016.07.274).

[12] S. Khamayseh and A. Halawani, "Cooperative Spectrum Sensing in Cognitive Radio Networks: A Survey on Machine Learning-based Methods", *Journal of Telecommunication and Information Technology*, vol. 2020, no. 3, pp. 36–46, 2020 (https://doi.org/10.26636/jtit.2020.137219).

[13] F.Z. Fagroud *et al.*, "Connected Devices Classification Using Feature Selection with Machine Learning", *IAENG International Journal of Computer Science*, vol. 49, no. 2, pp. 445–452, 2022 (https://www.iaeng.org/IJCS/issues_v49/issue_2/IJCS_49_2_18.pdf).

[14] A. Rawat, R.N. Yadav, and S.C. Shrivastava. "Neural Network Applications in Smart Antenna Arrays: A Review", *AEU-International Journal of Electronics and Communications*, vol. 66, no. 11, pp. 903–912, 2012 (https://doi.org/10.1016/j.aeue.2012.03.012).

[15] B. Hamdi, S. Limam, and T. Aguili, "Uniform and Concentric Circular Antenna Arrays Synthesis for Smart Antenna Systems Using Artificial Neural Network Algorithm", *Progress In Electromagnetics Research B*, vol. 67, pp. 91–105, 2016 (https://doi.org/10.2528/PIERB16031508).

[16] F. Andriulli, P.-Y. Chen, D. Erricolo, and J.-M. Jin, "Guest Editorial. Machine Learning in Antenna Design, Modeling, and Measurements", *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 7, pp. 4948–4952, 2022 (https://doi.org/10.1109/TAP.2022.3189963).

[17] D. Shi, C. Lian, K. Cui, Y. Chen and X. Liu, "An Intelligent Antenna Synthesis Method Based on Machine Learning", *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 7, pp. 4965–4976, 2022 (https://doi.org/10.1109/TAP.2022.3182693).

[18] A. Ibrahim, H.F. Abutarboush, A. Wagdy, M.M. Fouad, and E-S.M. Elkenawy, "An Optimized Ensemble Model for Prediction the Bandwidth of Metamaterial Antenna", *Computers, Materials & Continua*, vol. 71, no. 1, pp. 199–213, 2022 (https://doi.org/10.32604/cmc.2022.021886).

[19] Y. Anzai, *Pattern Recognition and Machine Learning*, Elsevier, 2012 (ISBN: 9780080513638).

[20] I.H. Sarker, "Machine Learning: Algorithms, Real-world Applications and Research Directions", *SN Computer Science*, vol. 2, no. 3, pp. 1–21, 2021 (https://doi.org/10.1007/s42979-021-00592-x).

[21] S.S. Keerthi, S.K. Shevade, C. Bhattacharyya, K.R.K. Murthy, "Improvements to Platt's SMO Algorithm for SVM Classifier Design", *Neural Computation*, vol. 13, no. 3, pp. 637–649, 2001 (https://doi.org/10.1162/089976601300014493).

[22] J. Platt, "Sequential Minimal Optimization: A Fast Algorithm for Training Support Vector Machines", *Microsoft Technical Report MSR-TR-98-14*, 1998 (https://www.microsoft.com/en-us/research/wp-content/uploads/2016/02/tr-98-14.doc).

[23] S. Haykin, *Neural Network and Machine Learning*, 3rd ed., Pearson International, 936 p., 2009 (ISBN: 9780131471399).

[24] Y. Han, Y. Tao, W. Zhang, W. Cui, and T. Shi, "Perceptron Neural Network Image Encryption Algorithm Based on Chaotic System", *IAENG International Journal of Computer Science*, vol. 50, no.1, pp. 42–50, 2023 (https://www.iaeng.org/IJCS/issues_v50/issue_1/IJCS_50_1_06.pdf).

[25] Y. Ye, J. Che, and H. Wang, "Optimal Component IGSCV-SVR Ensemble Model Improved by VMD for Ultra-short-term Wind Speed Forecasting", *Engineering Letters, IAENG*, vol. 30, no. 3, pp. 1166–1175, 2022 (https://www.engineeringletters.com/issues_v30/issue_3/EL_30_3_29.pdf).

[26] Y. Yang, *Temporal Data Mining via Unsupervised Ensemble Learning*, Elsevier, USA, 172 p., 2016 (ISBN: 9780128116548).

[27] M.S. Alajmi and A.M. Almeshal, "Least Squares Boosting Ensemble and Quantum-behaved Particle Swarm Optimization for Predicting the Surface Roughness in Face Milling Process of Aluminum Material", *Applied Sciences*, vol. 11, no. 5, art. no. 2126, 2021 (https://doi.org/10.3390/app11052126).

[28] J.R. Quinlan, "Induction of Decision Trees", *Machine Learning*, vol. 1, no. 1, pp. 81–106, 1986 (https://doi.org/10.1007/BF00116251).

[29] B.T. Jijo and A.M. Abdulazeez, "Classification Based on Decision Tree Algorithm for Machine Learning", *Journal of Applied Science and Technology Trends*, vol. 2, no. 1, pp. 20–28, 2021 (https://doi.org/10.38094/jastt20165).

[30] Y. Yang, I.G. Morillo, and T.M. Hospedales, "Deep Neural Decision Trees", *2018 ICML Workshop on Human Interpretability in Machine Learning (WHI 2018)*, Stockholm, Sweden, 2018 (https://doi.org/10.48550/arXiv.1806.06988).

[31] Y.-S. Lau, Z.M. Hussain, and R.J. Harris, "A Weight-vector LMS Algorithm for Adaptive Beamforming", *IEEE Region 10 Conference TENCON*, Chiang Mai, Thailand, pp. 494–498, 2004 (https://doi.org/10.1109/TENCON.2004.1414465).

[32] H. Che, C. Li, X. He, and T. Huang, "A Recurrent Neural Network for Adaptive Beamforming and Array Correction", *Neural Networks*, vol. 80, pp. 110–117, 2016 (https://doi.org/10.1016/j.neunet.2016.04.010).

[33] I. Mallioras, Z.D. Zaharis, P.I. Lazaridis, and S. Pantelopoulos, "A novel realistic approach of adaptive beamforming based on deep neural networks", *IEEE Transactions on Antennas and Propagation*, vol. 70, no. 10, pp. 8833–8848, 2022 (https://doi.org/10.1109/TAP.2022.3168708).

[34] A.H. Sallomi and S. Ahmed, "Elman Recurrent Neural Network Application in Adaptive Beamforming of Smart Antenna System", *International Journal of Computer Applications (IJCA)*, vol. 129, no. 11, pp. 38–43, 2015 (https://doi.org/10.5120/ijca2015907041).

[35] F. Liu, Y. Wu, H. Duan, and R. Du, "SVR-CMT Algorithm for Null Broadening and Sidelobe Control", *Progress In Electromagnetics Research*, vol. 163, pp. 39–50, 2018 (https://doi.org/10.2528/PIER18061106).

**Barsa Samantaray, M.Sc.**
ECE Department, ITER
https://orcid.org/0000-0001-7184-2123
E-mail: barsasamantaray@soa.ac.in
SOA University, Bhubaneswar, India
https://soa.ac.in

**Kunal Kumar Das, Ph.D.**
ECE Department, ITER
E-mail: kunaldas@soa.ac.in
SOA University, Bhubaneswar, India
https://soa.ac.in

**Jibendu Sekhar Roy, Ph.D.**
School of Electronics Engineering
https://orcid.org/0000-0002-3571-2708
E-mail: drjsroy@rediffmail.com
KIIT University, Bhubaneswar, Odisha, India
https://kiit.ac.in