


Optimization of Predicting Harvested Power of Toroidal Electromagnetic Energy Harvesters Using ABC Algorithm


Kadir Ileri

(Bandirma Onyedil Eylül University, Electrical and Electronics Engineering, Bandirma, Balıkesir, Türkiye; Centre for Future Transport and Cities, Coventry University, CV1 5FB, Coventry, United Kingdom,

 <https://orcid.org/0000-0002-5041-6165>, kileri@bandirma.edu.tr, ae5198@coventry.ac.uk)


M. Şamil Balci

(Bandirma Onyedil Eylül University, Electrical and Electronics Engineering, Bandirma, Balıkesir, Türkiye

 <https://orcid.org/0000-0002-3926-253X>, mbalci@bandirma.edu.tr)

Adem Dalcalı

(Bandirma Onyedil Eylül University, Electrical and Electronics Engineering, Bandirma, Balıkesir, Türkiye

 <https://orcid.org/0000-0002-9940-0471>, adalcali@bandirma.edu.tr)

Abstract: Energy harvesting is an effective solution, especially in scenarios with low power requirements, using sources such as magnetic fields, vibrations, and wind. This study focuses on predicting harvested power of toroidal electromagnetic energy harvesters using various machine learning methods, including Least Absolute Shrinkage and Selection Operator (Lasso), Adaptive Boosting (AdaBoost), K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest, Gradient Boosting Machine (GBM), Extreme Gradient Boosting (XGBoost), and Light Gradient Boosting Machine (LightGBM). To enhance the performance of these models, Artificial Bee Colony (ABC) optimization has been applied. The experiments were conducted using 1,300 trials across seven toroidal cores with varying sizes and magnetic permeabilities. During each experiment, the line current was varied between 0–100 A, and the resulting induced voltage and current were recorded. These measurements were used to create a comprehensive dataset named the Toroidal-Energy-Harvesting Dataset, enabling accurate power prediction. The performance of the machine learning models was assessed using statistical metrics, including R^2 , MSE, MAE, and RMSE. Among the evaluated models, the ABC-optimized XGBoost (ABC-XGBoost) demonstrated the highest performance, achieving an R^2 value of 0.9993, an MSE of 247.1, an MAE of 9.8, and an RMSE of 15.7, indicating superior accuracy and minimal error. The comparative analysis clearly shows that proposed ABC-XGBoost outperformed the other models, making it the most effective solution for accurate power prediction in the Toroidal-Energy-Harvesting Dataset.

Keywords: Energy harvesting, Optimization, Artificial Bee Colony, Random Forest, XGBoost, LightGBM

Categories: H.4, J.2, J.7

DOI: 10.3897/jucs.140979

1 Introduction

The demand for energy is rapidly increasing with the development of technology and industry. Ensuring a continuous balance between supply and demand is a challenging task for the energy sector. Energy supply security can be achieved by diversifying production sources and using resources efficiently. The environmental damage caused by fossil fuel sources and the limited known reserves have particularly directed the energy sector towards alternative sources. Alternative energy sources like solar, wind, or hydraulic power have drawbacks such as long construction times and high installation costs. In situations where the power grid is unavailable and the installation of sources like solar or hydraulic power is inefficient and costly, energy harvesting is an alternative option for meeting the power needs of low-power devices like sensors [Balci and Dalcalı, 2023].

Energy harvesting involves obtaining small amounts of energy from environmental sources (such as wind, magnetic fields, vibrations, etc.) using mechanisms like piezoelectric and electromagnetic systems. Among these mechanisms, electromagnetic energy harvesting stands out due to its high energy density, independence from weather conditions, robust structure, and simple installation advantages [Gao et al., 2024, Ye et al., 2024, Veney and D'Souza, 2023, Yin et al., 2024]. Electromagnetic harvesters consist of a core and one or more coils where the voltage is induced. In this mechanism, the current-carrying conductor forms the primary winding of the transformer, while the coils on the core form the secondary winding. The harvested voltage, and consequently the power, is influenced by factors such as the core material, line current, and core dimensions [Ye et al., 2024, Monagle et al., 2023].

It is important to know the voltage and harvested power in harvester systems. In this way, the performance of the harvesters under varying conditions can be obtained. Performance values can be obtained through analytical methods, computer-aided programs, or experimentally. Techniques like finite element analysis can test many combinations using computer-aided methods. However, these analyses require a long time. Experimental studies, on the other hand, require significant effort and financial resources. Therefore, using machine learning methods can obtain the performance of harvesters without the disadvantages mentioned above [Yuan et al., 2023].

There are many studies in the literature that examine the performance of energy harvesters [Hoang and Ebeid, 2023, Gao et al., 2023, Lin et al., 2021]. In the study examining the performance of a piezoelectric energy harvester, the impact of input variables on the root mean square value of the harvested voltage has been obtained using machine learning methods such as Decision Tree Regressor (DTR), Random Forest, and Gradient Boosting Regression Trees (GBRT) [Zhang et al., 2021]. In a similar study, the power and voltage values of a piezoelectric harvester have been examined experimentally. By varying the system frequency in the experiments, the changes in voltage and power have been estimated. The data obtained experimentally were both validated and predicted using a hybrid Deep Belief Network with Sandpiper Optimization Algorithm [Bhosale et al., 2023].

In another study, Li and Han address the development of an ARIMA-based model that takes into account seasonal variations to predict energy harvested from solar power. The Adaptive Seasonal Auto-Regressive Integrated Moving Average (ASARIMA) model is dynamically adjusted by leveraging the similarity in weather conditions across

different days, and its training is optimized based on these similarities. Experimental results reveal that the ASARIMA model outperforms other existing energy prediction algorithms. The ASARIMA model is compared with traditional forecasting algorithms such as Exponentially Weighted Moving Average (EWMA), Weather-Conditioned Moving Average (WCMA), and Universal Dynamic WCMA (UD-WCMA). ASARIMA exhibited a lower error rate compared to other algorithms, particularly under stable weather conditions [Li and Han, 2022]. Cammarano et al. aim to develop online energy prediction models for energy harvesting in environmentally powered wireless sensor networks. The proposed models predict future energy intake using past energy observations from solar and wind sources [Cammarano et al., 2016]. In the study where a new prediction model for piezoelectric energy harvesting is developed, the application of the proposed method in energy management for industrial wireless sensors is discussed. The proposed model predicts energy based on vibration data and uses this prediction for the energy management of wireless sensors. It analyzes vibration data obtained from past datasets and predicts future energy production levels based on these data [Mouapi et al., 2020]. In experiments conducted in a wind tunnel, the output voltage of a piezoelectric harvester was similarly predicted using machine learning methods. The effects of installation angles and coverages of finshape attachments on the output voltage were successfully estimated using DTR, Random Forest, and GBRT methods [Song et al., 2023].

Conventional modelling techniques for electromagnetic energy harvesters typically depend on analytical models or numerical simulations like the finite element method [Huang et al., 2021]. Although these approaches provide valuable physical understanding, they are computationally intensive, demand accurate material properties, and face challenges when dealing with intricate geometries or dynamic environments. Alternatively, experimental approaches are often costly and time-consuming, especially when evaluating a broad spectrum of design variables. In recent years, data-driven approaches like machine learning have emerged as promising alternatives because they can model nonlinear systems directly from experimental data without the need for explicit physical modelling. Therefore, machine learning methods can predict the performance of energy harvesters while avoiding the drawbacks associated with traditional modelling and experimental techniques [Balcı and Dalcalı, 2024, Yuan et al., 2023]. Moreover, as highlighted by Riba et al., electric field energy harvesting systems present additional modelling challenges due to complex capacitive coupling phenomena. Their study demonstrates that some key parameters such as stray capacitances cannot be derived analytically and require finite element simulations for accurate estimation. This further limits the generalizability and scalability of traditional models. These limitations underline the need for flexible and robust modelling frameworks [Riba et al., 2023]. In addition to the challenges associated with analytical modelling, recent studies have also highlighted practical limitations observed in experimental approaches. Sabat et al., conducted detailed laboratory and in-situ experiments on electromagnetic energy harvesting systems and emphasized the sensitivity of harvester performance to factors such as antenna type, polarization mismatches, frequency detuning, and environmental variability. Their findings indicate that even small deviations in antenna alignment or operating frequency can significantly alter output voltage levels, often falling below thresholds necessary to activate power conditioning circuits [Sabat et al., 2023]. These observations underscore the difficulty of developing generalizable empirical models for electromagnetic

harvesters and further motivate the use of machine learning-based predictive models that can capture such nonlinear dependencies under variable conditions. Furthermore, the impact of critical variables, such as line current, load, and winding current, on harvested power has not been sufficiently explored. This study addresses these gaps by creating a comprehensive dataset named the Toroidal-Energy-Harvesting Dataset, applying various optimized machine learning models, and conducting a detailed performance comparison to identify the most effective model for accurate power prediction in electromagnetic energy harvesting systems.

This study aims to accurately predict the harvested power of toroidal electromagnetic energy harvesters using a diverse set of machine learning techniques, including Least Absolute Shrinkage and Selection Operator (Lasso), Adaptive Boosting (AdaBoost), K-Nearest Neighbors (KNN), Decision Tree (DT), Random Forest, Gradient Boosting Machine (GBM), XGBoost, and LightGBM. To further enhance the predictive capabilities of these models, ABC optimization has been applied. The analysis begins with the creation of a comprehensive dataset, the Toroidal-Energy-Harvesting Dataset, derived from extensive experimental measurements. Among the evaluated models, the ABC-optimized XGBoost demonstrated the highest performance. This proposed ABC-XGBoost approach not only offers a robust solution for accurate power prediction in electromagnetic energy harvesters but also highlights the significant impact of optimization techniques in improving model performance. The main contributions of this article are as follows:

- A comprehensive controlled experimental setup was designed and implemented using seven distinct toroidal cores with varying sizes and magnetic permeabilities. From this setup, a detailed dataset named the Toroidal-Energy-Harvesting Dataset was created, providing a valuable resource for researchers to further their studies in electromagnetic energy harvesting and machine learning-based power prediction.
- Various machine learning models, including Lasso, AdaBoost, KNN, DT, Random Forest, GBM, XGBoost, and LightGBM, were employed to accurately predict the harvested power of toroidal-type electromagnetic energy harvesters.
- ABC optimization was applied to enhance the performance of these machine learning models. Among them, the ABC-optimized XGBoost model demonstrated the highest predictive accuracy and the lowest error rates, showcasing the effectiveness of the optimization approach.

The remainder of this paper is organized as follows. In Section 2, the details of experimental setup, and created dataset are provided. The theoretical background and the machine learning methods used in the study are detailed in Section 3. The experimental results are presented in the Section 4. Section 5 concludes the paper.

2 Experimental Setup and Toroidal-Energy-Harvesting Dataset

In the experimental study, the line current was increased from 0 to 100 A in increments of 5 A. To achieve the targeted values of line current, a 1.5 kVA transformer with 220V:12V conversion ratio, 50 Hz frequency was used, along with a rheostat which

serves as a preliminary resistor to finely adjust the line current. A clamp amperemeter has been placed to monitor the line current, while multimeters and an oscilloscope have been used to measure the induced voltage and the current. The experimental setup which has been arranged for this study is shown in Figure 1.

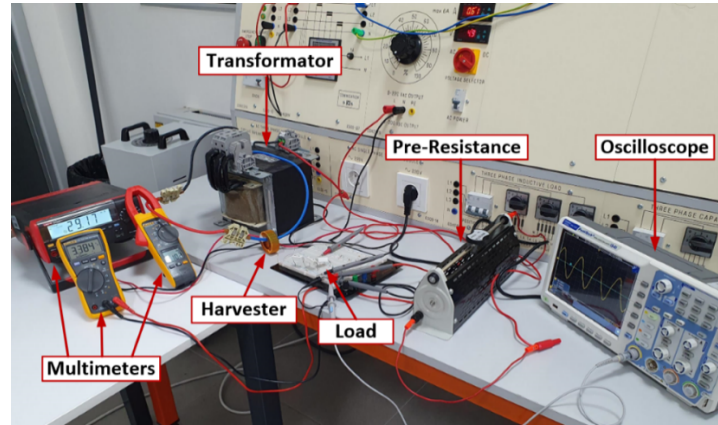


Figure 1: Experimental setup

Within the scope of the experiment, seven toroidal cores of varying sizes and magnetic permeabilities have been used, as illustrated in Figure 2. The numbers from 1 to 7 in Figure 2 correspond to the core numbers listed in Table 2. Detailed information about the core sizes and materials is provided in Table 1, where the core dimensions are specified as outer diameter, inner diameter, and height, respectively.



Figure 2: Toroidal cores used in the experiment

Core Number	Core Dimensions (mm)	Core Permeability (μ_r)
1	31.7 x 19 x 8	620
2	61 x 35.5 x 20	620
3	34 x 20.5 x 10	4300
4	50 x 28.5 x 20	4300
5	50 x 30 x 20	2100
6	60 x 30 x 20	2100
7	63 x 38 x 20	2100

Table 1: Specification of cores used in the experiment

Within a range of 0 to 100 A line current and under various loads a total of 1,300 experiments have been conducted with seven different cores. In these experiments the induced voltage and the current passing through the windings have been measured. Based on the measured values, the power obtained in each experimental stage has been calculated. As a result of these experiments, the Toroidal-Energy-Harvesting Dataset [Toroidal-Energy-Harvesting, 2025] was created, and a few examples from this dataset are provided in Table 2.

Core Number	Line Current (A)	Load (Ω)	V_{Load} (V)	$I_{Winding}$ (mA)	Power (mW)
1	5	10	0.077	6.82	0.52514
1	10	20	0.277	13.14	3.63978
1	15	30	0.599	19.41	11.62659
2	20	1.5	0.0738	38.95	2.87451
2	25	5	0.301	49.82	14.99582
2	30	10	0.628	58.8	36.9264
3	35	15	1.627	105.6	171.8112
3	40	20	2.011	97.5	196.0725
3	45	25	2.272	85.3	193.8016
4	50	15	3.176	210.6	668.8656
4	55	30	4.618	154.1	711.6338
5	60	30	5.365	179.2	961.408
5	65	50	6.88	139.6	960.448
5	70	75	8.04	107.8	866.712
6	75	15	4.166	298.6	1243.9676
6	80	30	7.67	263.1	2017.977
6	85	40	8.82	233.1	2055.942
7	90	25	6.181	252.6	1561.3206
7	95	35	8.3	242.5	2012.75
7	100	60	10.66	183.3	1953.978

Table 2: Sample records from the created Toroidal-Energy-Harvesting Dataset

The first column of the data presented in Table 2 indicates the core number. The second column shows the line current, and the third column shows the value of the resistive load connected to the harvester. The fourth column indicates the induced voltage, the fifth column represents the current drawn by the load. Finally, the harvested power is provided in the last column.

3 Theoretical Background and Methods

Induced voltage calculation in harvester

Electromagnetic energy harvesters are specifically designed to capture energy from the magnetic fields generated by current flowing through conductors. The current flowing through a conductor creates a magnetic flux density around as shown in Equation 1 [Balci and Dalcalı, 2023].

$$\vec{B} = \frac{\mu i(t)}{2\pi r} \quad (1)$$

In the equation, μ represents the permeability of the material, i indicates the line current and r indicates the radius of the core. This magnetic flux density allows the circulation of magnetic flux in the core. The magnetic flux is expressed by Equation 2 [Balci and Dalcalı, 2023].

$$\phi_{21} = \int \vec{B} \vec{d}s \quad (2)$$

This magnetic flux allows voltage to be induced in the coils as in Equation 3 [Balci and Dalcalı, 2023].

$$V_2 = N_2 \frac{d\phi_{21}(t)}{dt} \quad (3)$$

For the toroidal core magnetic energy harvester, the induced voltage can be expressed by Equation 4 [Balci and Dalcalı, 2023].

$$V_2 = N_2 \mu f i(t) \sin(\omega t) \ln\left(\frac{b}{a}\right) h \quad (4)$$

As seen in Equation 4, the voltage is affected by the number of turns (N_2), the magnetic permeability of the core material (μ_r), the amount of current flowing through the conductor (i), the frequency (f), and the core size. The outer diameter of the core is represented by b , the inner diameter by a , and the height by h .

Machine learning methods

The selection of machine learning models in this study was guided by the objective of achieving a comprehensive performance comparison for accurate power prediction in toroidal-type electromagnetic energy harvesters. Specifically, Lasso, AdaBoost, KNN, DT, Random Forest, GBM, XGBoost, and LightGBM were chosen due to their diverse

algorithmic characteristics. Lasso represents a linear model, KNN is an instance-based method, while DT and Random Forest are tree-based models. The remaining models such as AdaBoost, GBM, XGBoost, and LightGBM are powerful ensemble learning techniques known for their ability to capture complex nonlinear relationships in data. This diverse selection not only ensures a thorough evaluation of model performance but also aligns with existing literature, where these models are well-established for regression tasks. Such a diverse approach enables a robust assessment, allowing the identification of the most suitable model for accurate power prediction.

3.2.1. KNN

K-Nearest Neighbor (KNN) is widely used machine learning algorithm that employed for both regression and classification processes. It involves a parameter called k , which represents the number of nearest neighbors [Uddin et al., 2022, Xiong and Yao, 2021].

The KNN method operates by identifying the number of k nearest neighbors from a given dataset in relation to a given query point. These nearest neighbors are identified based on their closest distances from the query point. The most commonly used distance metric for this calculation is the Euclidean distance. This distance can be computed using Equation 5. After the algorithm locates the k nearest neighbors, it applies the majority voting rule to determine the most prevalent class. On the other hand, for regression tasks, it applies the rule of taking the mean of the k nearest neighbors.

$$D(x_i, x_q) = \sqrt{\sum_{l=1}^n |x_i^{(l)} - x_q^{(l)}|^2} \quad (5)$$

where x_i , and x_q are the data points and the query point, respectively.

3.2.2. Lasso

Lasso regression, which stands for Least Absolute Shrinkage and Selection Operator, is a regularization (shrinkage) method [Wang et al., 2024]. It shares this characteristic with other methods as they aim to minimize the sum of squared residuals (RSS) alongside a penalty term. By introducing a penalty term that penalizes the absolute size of coefficients, Lasso regression encourages certain coefficients to be precisely zero. This attribute renders Lasso regression particularly beneficial for high-dimensional datasets [Qian et al., 2020]. The goal of Lasso regression is to minimize the following objective function [Ng and Newton, 2022]:

$$L = RSS + \lambda \sum_{j=1}^p |\beta_j| \quad (6)$$

where λ is the shrinkage parameter, p is the number of estimators, and β_j are the regression coefficients. The RSS, which presents the difference between the estimated and actual values, can be obtained as follows:

$$RSS = \sum (y - \hat{y})^2 \quad (7)$$

3.2.3. Decision tree

Decision Tree (DT) is a crucial algorithm extensively utilized for data analysis, capable of handling both regression and classification tasks [Pekel, 2020]. It comprises root node, internal nodes, edges, and leaf nodes.

The fundamental elements of a DT are pointed out as follows [Kushwah et al., 2022]. Firstly, there are nodes, which serve as positions where the tree divides based on the values of independent variables from the dataset. Secondly, edges denote the direction of the divisions. They guide the decision process from one node to another. The root represents the initial node where the first split occurs. It sets the foundation for the subsequent divisions. Finally, leaf nodes (also referred to as terminal nodes) provide the ultimate predictions or outcomes of the tree.

The underlying mechanism of DT algorithm involves generating a decision tree based on provided instances. The objective of this method is to discover the optimal tree by keeping the fitness function at minimum. The learning process begins with fitting a regression model to the output variable using each input variable. Then, the dataset is divided at numerous division points for each input variable and the method computes the errors between the estimated and actual values at each division. Finally, the variable resulting in the lowest error according to the predefined fitness function is preferred as the division point. This iterative procedure continues recursively.

3.2.4. AdaBoost

AdaBoost, stands for Adaptive Boosting, is one of the widely used boosting methods introduced by Freund and Schapire [Freund and Schapire, 1996]. It is utilized for both classification and regression tasks.

In AdaBoost regression, weak regression models are often straightforward, like decision trees with limited depth. The process involves iteratively training these weak models on the dataset while adjusting the weights of data points according to the errors of previous models. With each iteration, AdaBoost regression assigns greater weight to data points that were inaccurately predicted by prior weak models. This allows following weak models to concentrate on challenging instances, thereby enhancing the overall regression performance over time. Finally, the predictions from all weak regression models are aggregated to derive the final regression prediction [Ileberi et al., 2021]. This final prediction can be obtained as follows:

$$Y = \sum_{m=1}^M c_m y_m \quad (8)$$

where M represents the total number of weak regression models, y_m is the prediction of the m^{th} weak model and c_m is the coefficient assigned to the m^{th} weak model.

3.2.5. Random forest

Random Forest is proposed by Breiman in 2001 [Breiman, 2001]. It is an effective machine learning method that uses a number of DTs. One of the disadvantages of Random Forest is that it may not operate well on unbalanced datasets. On the contrary, Random Forest is resistant to overfitting and has ability to handle large datasets with high dimensionality [Maruf et al., 2016].

As illustrated in Figure 3, the Random Forest is an ensemble model that consists of N decision trees. Each decision tree gets independent inputs (x_i) and produces separate outputs estimations (y_i), where $i = 1, \dots, N$. These estimations made by decision trees are aggregated to generate the final estimation. While the final estimation (Y) is the class estimated by the majority trees for the classification case, it is the average of decision trees' outputs for the regression case which is given in Equation 9.

$$Y = \frac{1}{N} \sum_{i=1}^N y_i \quad (9)$$

Each decision tree in the forest grows using different parts of the training dataset obtained as a result of the bagging process, which leads the diversity between decision trees to prevent overfitting. The bagging involves creating decision trees by resampling randomly the original dataset with replacement.

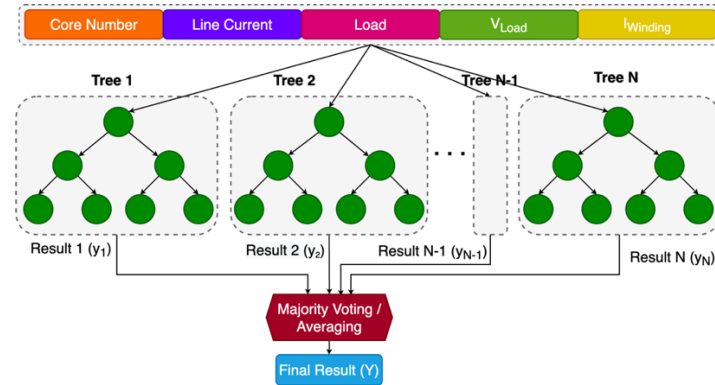


Figure 3: Structure of ensemble Random Forest method

3.2.6. GBM

Gradient Boosting Machine (GBM) is a widely used ensemble learning technique, was initially proposed by Friedman [Friedman, 2001]. Since its introduction, GBM has proven to be an effective method for solving both regression and classification problems. It constructs a strong predictive model by sequentially combining multiple weak learners, typically decision trees. Each subsequent tree is trained to correct the residual errors of the previous model, effectively minimizing a predefined loss function using gradient descent optimization.

GBM is highly versatile, allowing the use of various loss functions such as Mean Squared Error, Logarithmic Loss, and Huber Loss [Ferreira, 2022]. This flexibility

enables GBM to be effectively applied to a wide range of machine learning problems. The general form of the GBM model can be expressed as:

$$Y = \sum_{m=1}^M y_m h_m \quad (10)$$

where Y is the final model, M is the number of boosting rounds, y_m is the contribution of the m -th weak learner, and h_m is the weak learner at the m -th iteration. The weak learners are trained to minimize the loss function.

3.2.7. XGBoost

Extreme Gradient Boosting (XGBoost), which is proposed by Tianqi Chen and Carlos Guestrin [Chen and Guestrin, 2016], is an advanced implementation of gradient boosting that enhances performance and efficiency. It introduces several enhancements over traditional GBM, including a more efficient tree learning algorithm, built-in regularization to prevent overfitting, and parallel processing capabilities.

The XGBoost model can be represented as a sum of K decision trees:

$$F = \sum_{k=1}^K f_k \quad (11)$$

where F is the final model prediction, and f_k is the prediction of the k -th decision tree. The objective function that XGBoost minimizes is defined as:

$$L(\theta) = \sum_{i=1}^N l(y_i, \hat{y}_i) + \sum_{k=1}^K \Omega(f_k) \quad (12)$$

where N is the number of training samples, $l(y_i, \hat{y}_i)$ is the loss function, measuring the difference between the true value y_i and the predicted value \hat{y}_i , and $\Omega(f_k)$ is the regularization term for each tree f_k , which controls model complexity and defined as follows:

$$\Omega(f_k) = \gamma T + 0.5 \cdot \lambda \sum_{j=1}^T w_j^2 \quad (13)$$

where T is the number of leaves in the tree, w is the score on the j -th leaf, γ is the penalty for the number of leaves, and λ is the L_2 regularization term on the leaf scores, which regulates the complexity of the tree structure.

3.2.8. LightGBM

Light Gradient Boosting Machine (LightGBM) is a gradient boosting framework designed to handle large datasets with fast training and low memory consumption [Ke et al., 2017]. Unlike traditional gradient boosting methods that use level-wise tree

growth, LightGBM employs a leaf-wise growth strategy, where it selects and splits the leaf with the highest split gain among all current leaves. While this approach allows for deeper trees and improved model accuracy, it also increases the risk of overfitting. To mitigate this, LightGBM introduces a maximum depth limit, ensuring high efficiency and preventing excessive complexity.

LightGBM enhances computational efficiency through two key techniques: Gradient-Based One-Side Sampling (GOSS) and Exclusive Feature Bundling (EFB) [Ke et al., 2017]. GOSS focuses on data instances with large gradient values, which are more informative for model training, allowing LightGBM to achieve accurate split point determination with reduced data. EFB further accelerates training by bundling mutually exclusive features into a single feature, minimizing the number of features without losing critical information. These innovations enable LightGBM to achieve faster training and better performance, especially on large-scale datasets.

Similar to XGBoost, the final prediction of LightGBM is calculated as:

$$Y = \sum_{k=1}^K f_k \quad (14)$$

where K is the total number of decision trees in the model, and f_k is the prediction of the k -th tree.

Artificial bee colony optimization

ABC optimization, which stands for Artificial Bee Colony, is a population-based metaheuristic optimization algorithm inspired by the foraging behavior of honeybees. It mimics the food foraging process of honeybee colonies, consisting of three main types of bees: scout bees, employed bees, and onlooker bees [Karaboğa and Basturk, 2007].

Every employed bee investigates a solution within the neighborhood of its current location. Consequently, each employed bee adjusts its solution using the equation given as follows:

$$x'_{ij} = x_{ij} + \phi_{ij} \cdot (x_{ij} - x_{hj}) \quad (15)$$

where;

- x_{ij} is the current solution represented by i^{th} employed bee in the j^{th} dimension,
- x'_{ij} is the new solution,
- ϕ_{ij} is the random number that ranges from -1 to 1 ,
- h is different from i and determined randomly from 1 to total number of employed bees.

Onlooker bees pick a solution based on the information distributed by employed bees. The probability of picking a solution is determined by its fitness value. This probability value can be obtained as follows:

$$p_i = \frac{f_i}{\sum_{e=1}^E f_e} \quad (16)$$

where E is the number of solutions and f_i is the fitness value of the i^{th} employed bee's solution. If an employed bee's solution has not been enhanced for a certain number of iterations, it turns into a scout bee to explore new solutions. ABC optimization can be used for parameter optimization, feature selection, and model selection [Kuo et al., 2018]. In this study, the ABC is employed for optimizing the parameters of machine learning models.

4 Results and Discussion

In this study, the harvested power of toroidal type electromagnetic energy harvesters has been predicted using input variables such as core size parameters, line current, load, V_{Load} , and $I_{Winding}$. A variety of machine learning methods, including Lasso, AdaBoost KNN, DT, Random Forest, GBM, XGBoost, and LightGBM have been employed for this prediction task. These methods have further undergone an optimization process using the ABC approach to achieve their optimal configurations.

The overall workflow of this study is illustrated in Figure 4. The process began with data collection, where raw data were gathered from the harvester system. In the second phase, the input and output variables were defined, and the dataset was divided into training and testing sets, with 90% of the data allocated for training and 10% for testing. The third phase involved employing the machine learning models to predict the output variable (harvested power). Finally, in the fourth phase, model optimization was performed using the ABC algorithm, which was configured with 15 particles running for 50 iterations.

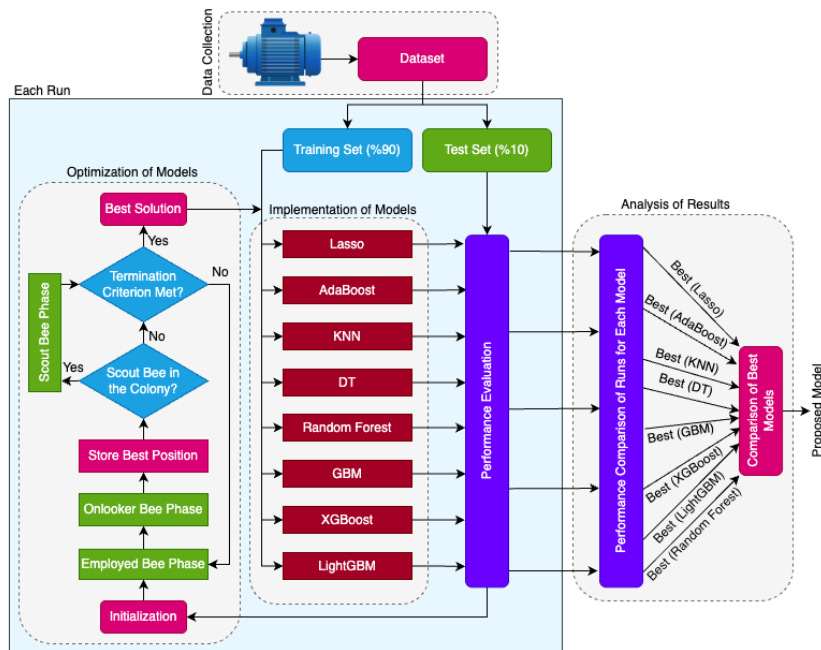


Figure 4: Flowchart of the processes conducted in this study

Cross-validation is a method utilized to evaluate the performance and generalization capability of a regression model. It entails dividing the dataset into several subsets, known as folds. The model is trained on one subset and validated on the remaining data. The mean of the performance results obtained from each iteration generates the final output. This iterative process provides a more accurate performance estimation, which helps prevent overfitting [Liu and Zhou, 2024]. In this study, 10-fold cross validation has been employed and the architecture of k-fold cross validation, which is general form of 10-fold, is illustrated in Figure 5. In addition to cross-validation, all processes have been repeated 10 times to ensure the robustness of the models.

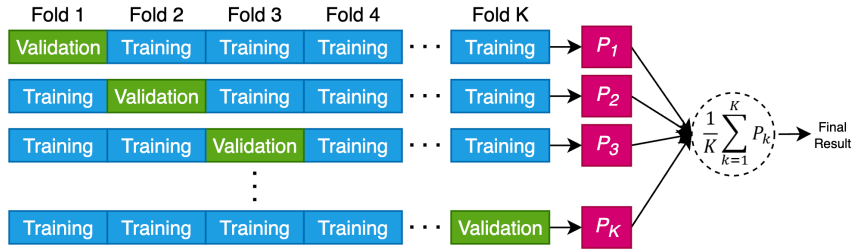


Figure 5: Architecture of k-fold cross validation

Table 3 presents the parameters for each machine learning model utilized in this study, along with their corresponding lower and upper bounds. These parameters were carefully selected to cover a reasonable range of values that could potentially yield optimal model performance. The optimization process, conducted using the ABC algorithm, explored these parameter ranges to identify the best configurations for each model. By adjusting the model parameters within the specified bounds, ABC ensured that the machine learning models were fine-tuned for maximum predictive accuracy, enhancing their overall performance in power prediction tasks.

Model	Parameter	Lower Limit	Upper Limit
Lasso	alpha	0.001	10
AdaBoost	n_estimators	50	250
	learning rate	0.01	1
KNN	n_neighbors	1	49
DT	max_depth	3	30
	min_samples_split	2	50
	min_samples_leaf	1	10
Random Forest	n_estimators	50	500
	max_depth	5	50
	min_samples_split	2	50
	min_samples_leaf	1	10
	min_weight_fraction_leaf	0	0.2
GBM	learning rate	0.1	0.8

	n_estimators	50	250
	max_depth	3	15
	min_samples_split	2	50
XGBoost	eta	0.01	0.3
	lambda	0	10
	alpha	0	10
	min_child_weight	1	10
	subsample	0.5	1
	n_estimators	50	250
	max_depth	3	15
LightGBM	num_leaves	10	500
	min_data_in_leaf	10	1000
	max_depth	3	30
	learning_rate	0.01	0.3

Table 3: Parameters of the machine learning model with their upper and lower limits

4.1 Regression evaluation metrics

The evaluation of the machine learning models has been performed by using four different statistical metrics: coefficient of determination (R^2), mean squared error (MSE), mean absolute error (MAE), and root mean squared error (RMSE). These metrics compute the deviation between actual and predicted values. A high value of the R^2 indicates high performance, while a high value in other metrics indicates low performance [Kumar et al., 2022]. These four metrics can be obtained as follows:

$$R^2 = \frac{\sum_{i=1}^s (O_i - \hat{O}_i)^2}{\sum_{i=1}^s (O_i - \bar{O})^2} \quad (17)$$

$$MSE = \frac{\sum_{i=1}^s (y_i - \hat{O}_i)^2}{s} \quad (18)$$

$$MAE = \frac{\sum_{i=1}^s |O_i - \hat{O}_i|}{s} \quad (19)$$

$$RMSE = \sqrt{\frac{\sum_{i=1}^s (O_i - \hat{O}_i)^2}{s}} \quad (20)$$

where O is the actual observation, \hat{O} is the estimated output, \bar{O} is the mean of actual observations, and s is the number of samples.

4.2 Experimental results

Figure 6 illustrates the accuracy convergence of the ABC optimization process for various models, including Lasso, AdaBoost, KNN, DT, Random Forest, GBM, XGBoost, and LightGBM over 50 iterations. XGBoost demonstrates the most rapid and consistent improvement in accuracy, starting from approximately 0.99 and quickly approaching 1.0. The rate of improvement slows after the 20th iteration, but it maintains the highest accuracy among all models. Random Forest initially exhibits relatively low accuracy. However, it significantly improves through ABC optimization, eventually achieving the second-best performance, just below XGBoost. LightGBM starts with lower accuracy compared to GBM, but the ABC optimization process allows it to surpass GBM, securing the third-best accuracy among the models. The accuracy of DT starts relatively high but shows minimal improvement after a slight increase, stabilizing early in the process. KNN and AdaBoost also show limited improvement, with their accuracies remaining mostly unchanged after a minor increase. Lastly, the accuracy of the Lasso accuracy is unaffected by the ABC optimization process, maintaining a nearly constant value throughout the iterations.

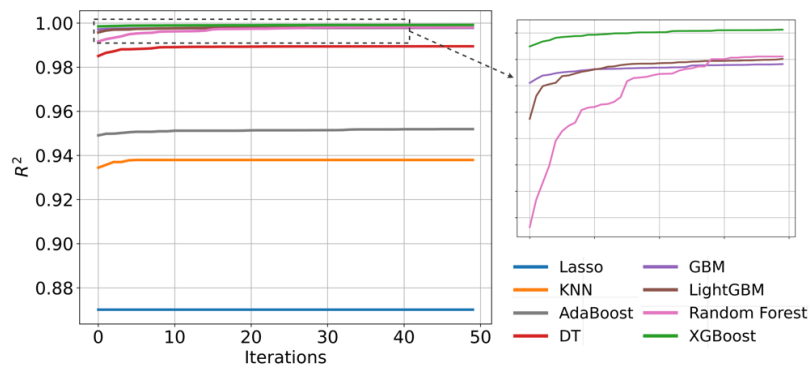


Figure 6: Convergence graph of the ABC optimization process

To ensure the robustness of the machine learning models under ABC optimization, all processes have been repeated 10 times and the accuracy results of these models at each run are given in Table 4. ABC-Lasso consistently achieves the same accuracy of 0.8701 across all runs, indicating complete stability but relatively lower predictive performance compared to the other models. This consistency is expected for Lasso, as it is a linear model with limited flexibility. Similarly, ABC-KNN maintains a nearly identical accuracy of 0.9381 across all runs, with only a slight decrease in the last run (0.9364). This slight variation suggests that the model is generally stable under ABC optimization but can be slightly sensitive to the initialization of the optimization process.

ABC-AdaBoost shows slight fluctuations in accuracy across the runs, ranging between 0.9504 and 0.9536. These variations can be attributed to the inherent randomness in the ensemble process and the effect of ABC optimization in fine-tuning the parameters of the model. ABC-DT maintains a high and stable accuracy of around 0.9895, with minimal fluctuations. This stability indicates that the decision tree model, once optimized, is highly consistent in its predictions.

For BC-GBM, the accuracy slightly varies between 0.9975 and 0.9981, which is expected for gradient boosting models because of their stochastic nature. ABC-LightGBM displays a similar pattern to ABC-GBM, with accuracies ranging between 0.9975 and 0.9985. The minor variations reflect the stochastic nature of the boosting process in LightGBM.

ABC-Random Forest demonstrates slightly larger variations in accuracy, ranging between 0.9972 and 0.9987. This is consistent with the inherent randomness of the bagging process in Random Forest, where different subsets of data can slightly affect the final model. Finally, ABC-XGBoost consistently achieves the highest accuracy among all models, ranging between 0.9989 and 0.9993. The small variations in accuracy suggest a high level of robustness, with the ABC optimization process effectively fine-tuning the model parameters for optimal performance.

In conclusion, the results indicate that the ensemble models (Random Forest, GBM, LightGBM, and XGBoost) generally achieve higher accuracies and exhibit strong stability with ABC optimization, while simpler models (Lasso, KNN, and AdaBoost) show comparatively lower accuracy. This performance difference can be attributed to the inherent characteristics of the models. Lasso, as a linear model, is limited in capturing complex nonlinear relationships, making it less suitable for the complex patterns in the data. KNN, an instance-based method, is sensitive to data distribution and suffers from high variance without advanced optimization. AdaBoost, although an ensemble method, relies on weak learners (typically decision trees with low depth) and may struggle with complex patterns. On the other hand, tree-based models (DT and Random Forest) can handle nonlinear relationships, and their ensemble variants (GBM, LightGBM, and XGBoost) further enhance performance by iteratively improving predictions, with XGBoost achieving the highest accuracy due to its robust gradient boosting mechanism and ABC optimization.

Model	Run 1	Run 2	Run 3	Run 4	Run 5
ABC-Lasso	0.8701	0.8701	0.8701	0.8701	0.8701
ABC-KNN	0.9381	0.9381	0.9381	0.9381	0.9381
ABC-AdaBoost	0.9536	0.9504	0.9510	0.9520	0.9519
ABC-DT	0.9895	0.9895	0.9894	0.9894	0.9893
ABC-GBM	0.9977	0.9980	0.9981	0.9978	0.9981
ABC-LightGBM	0.9982	0.9975	0.9981	0.9985	0.9980
ABC-Random Forest	0.9984	0.9984	0.9972	0.9987	0.9985
ABC-XGBoost	0.9990	0.9992	0.9993	0.9989	0.9992
Model	Run 6	Run 7	Run 8	Run 9	Run 10
ABC-Lasso	0.8701	0.8701	0.8701	0.8701	0.8701
ABC-KNN	0.9381	0.9381	0.9381	0.9381	0.9364
ABC-AdaBoost	0.9529	0.9510	0.9524	0.9521	0.9522
ABC-DT	0.9895	0.9895	0.9896	0.9896	0.9894
ABC-GBM	0.9980	0.9976	0.9976	0.9981	0.9975
ABC-LightGBM	0.9981	0.9981	0.9981	0.9978	0.9981
ABC-Random Forest	0.9984	0.9975	0.9977	0.9985	0.9980
ABC-XGBoost	0.9992	0.9992	0.9992	0.9991	0.9991

Table 4: Accuracy values of the machine learning models for each run

Figure 7 presents the mean and maximum accuracy values of the models across 10 runs. ABC-XGBoost achieved the highest performance, with both the highest mean and maximum accuracy, highlighting its superior optimization capability. ABC-LightGBM, ABC-Random Forest and ABC-GBM also performed well, closely following ABC-XGBoost. ABC-DT showed competitive performance, while ABC-AdaBoost and ABC-KNN demonstrated moderate accuracy. ABC-Lasso exhibited the lowest performance, indicating their limited ability to capture complex data relationships.

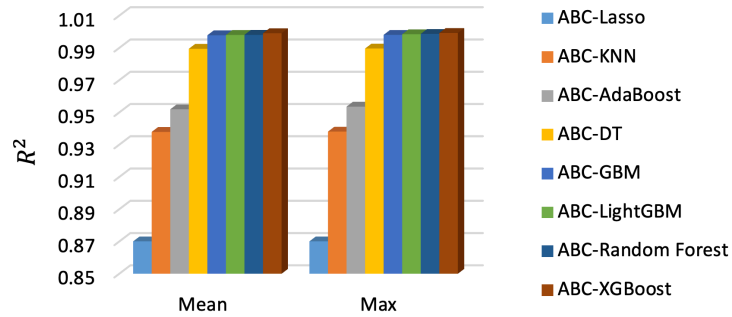


Figure 7: Mean and maximum accuracy values of the machine learning models

In addition to R^2 , the performance evaluation of the models based on MSE, MAE, and RMSE metrics reveals significant differences in their predictive capabilities. Table 5 provides a detailed comparison of the models across these metrics. Among all models, ABC-XGBoost demonstrates superior performance, achieving the highest R^2 value of 0.9993, which indicates an almost perfect correlation between the predicted and actual values. It also exhibits the lowest values for MSE (247.1), MAE (9.8), and RMSE (15.7), reflecting its exceptional ability to make accurate predictions with minimal error.

In contrast, ABC-Lasso shows the lowest predictive performance, with an R^2 value of 0.8701, alongside the highest MSE (35655.4), MAE (134.1), and RMSE (188.8). This result suggests that ABC-Lasso struggles to capture the underlying relationships in the data effectively. Other models, such as ABC-GBM, ABC-LightGBM, and ABC-Random Forest, also achieve high R^2 values, demonstrating their strong predictive power. However, their error metrics are slightly higher than those of ABC-XGBoost, indicating relatively less accurate predictions.

These findings highlight the effectiveness of the ABC-XGBoost model in achieving a high degree of goodness of fit while maintaining low prediction errors, making it the most reliable model for prediction of harvested power of toroidal electromagnetic energy harvesters.

*Best results are in bold.

Model	R ²	MSE	MAE	RMSE
ABC-Lasso	0.8701	35655.4	134.1	188.8
ABC-KNN	0.9381	16995.9	44.0	130.4
ABC-AdaBoost	0.9536	13246.7	93.0	115.1
ABC-DT	0.9896	2606.1	31.7	51.0
ABC-GBM	0.9981	549.1	14.6	23.4
ABC-LightGBM	0.9985	408.8	14.5	20.2
ABC-Random Forest	0.9987	419.5	11.7	20.5
ABC-XGBoost	0.9993	247.1	9.8	15.7

Table 5: Performance comparison of the machine learning models

Scatterplots of actual and predicted values provide a visual representation of how well the models are performing in predicting the output variable. In an ideal scenario, the points on the scatterplot would lie along a diagonal line, pointing to a perfect prediction where the estimated values match the actual values exactly. Deviations from this diagonal line point to discrepancies between the actual and estimated values [Suzuki et al., 2016]. The scatterplots of the models are illustrated in Figure 8. As seen clearly, the scatterplot with the most dispersion belongs to ABC-Lasso, while the scatterplot with the least dispersion belongs to ABC-XGBoost. This indicates that ABC-XGBoost exhibits the best performance, while ABC-Lasso demonstrates the least favorable performance. A closer analysis shows that the ABC-GBM, ABC-LightGBM, ABC-Random Forest, and ABC-XGBoost models provide highly accurate predictions, especially in the lower power range (0–500). Their predictive performance slightly decreases between 500 and 1,000, and further declines at values above 1,000, where prediction errors become more apparent. Conversely, the ABC-Lasso, ABC-KNN, ABC-AdaBoost, and ABC-DT models perform reasonably well at lower power levels (0–250) but exhibit significantly higher scatter at higher power values, indicating a decline in prediction accuracy. In conclusion, these findings suggest that while all models are effective at predicting lower power values, their accuracy generally decreases as the power increases.

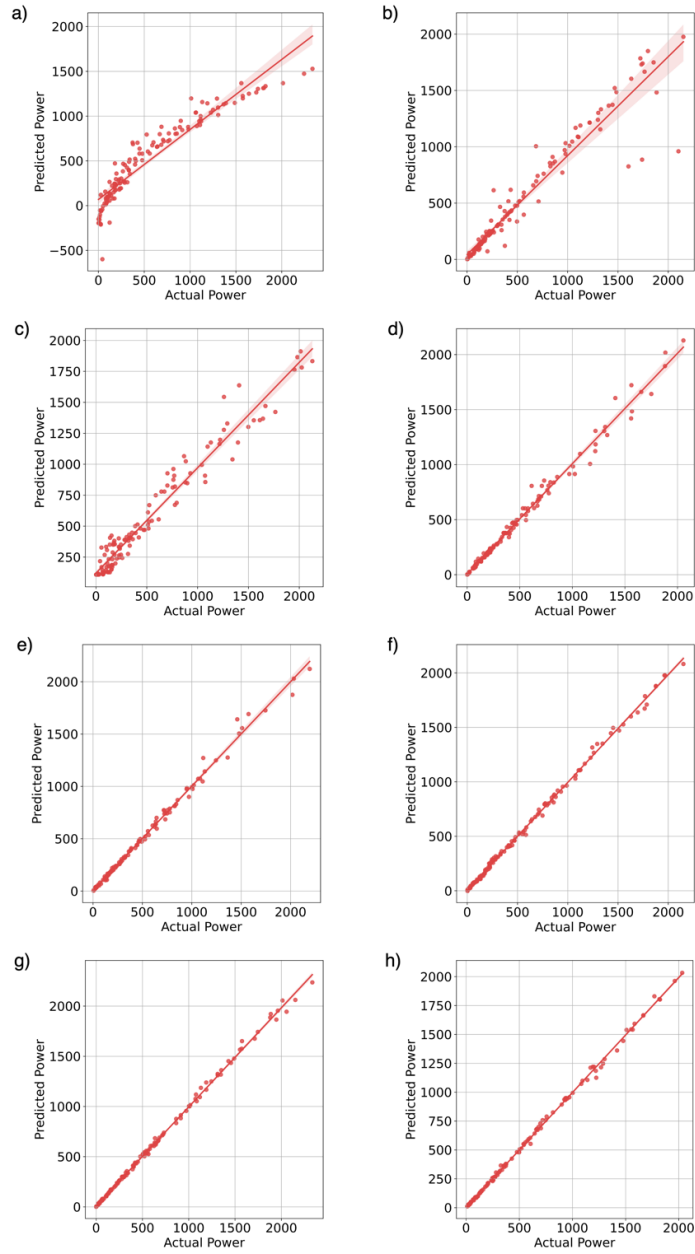


Figure 8: Scatterplots of actual and predicted values, a) ABC-Lasso, b) ABC-KNN, c) ABC-AdaBoost, d) ABC-DT, e) ABC-GBM, f) ABC-LightGBM, g) ABC-Random Forest, h) ABC-XGBoost

Figure 9 illustrates the swarm size analysis for the proposed ABC-XGBoost model, where the impact of swarm size on model performance is assessed using the R^2 metric.

The graph displays two curves: one representing the maximum R^2 values (Max) achieved at each swarm size and the other showing the mean R^2 values (Mean). Notably, as the swarm size increases from 5 to 15, there is a significant improvement in both the maximum and mean R^2 values, indicating enhanced model accuracy. After reaching a swarm size of 15, the R^2 values stabilize, with only minor variations observed beyond this point. This indicates that a swarm size of around 15 is sufficient to achieve optimal model performance, providing a balance between computational efficiency and model accuracy. The results clearly demonstrate that an appropriate swarm size is crucial for maximizing the predictive capability of the ABC-XGBoost model.

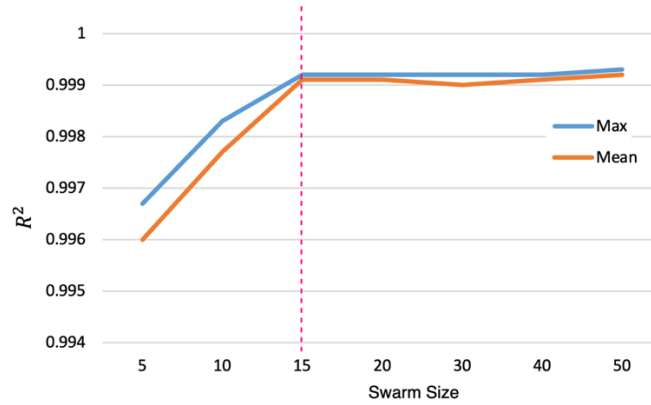


Figure 9: Analysis of swarm size in the ABC algorithm

4.3 Feature importance analysis

Feature importance analysis provides valuable insights into the contribution of each input variable toward the model's predictive performance. The results, illustrated in Figure 10, reveal significant variations in feature importance across different models. Specifically, $I_{Winding}$ is identified as the most influential feature for the ABC-AdaBoost and ABC-LightGBM models, while V_{Load} is the most critical feature for the remaining models, underscoring its strong impact on the harvested power prediction of the electromagnetic energy harvester. The third most influential feature differs among models. For instance, core number is the third most important feature for the ABC-XGBoost model, whereas line current holds this position for all other models. In contrast, the core number, load, and line current features generally exhibit lower importance in all models, indicating a relatively minor impact on the prediction results. These variations underscore that different machine learning models prioritize features differently, and selecting the most appropriate model can significantly impact prediction accuracy. The analysis also confirms that V_{Load} and $I_{Winding}$ are generally the most influential features for accurate power prediction in toroidal electromagnetic energy harvesters.

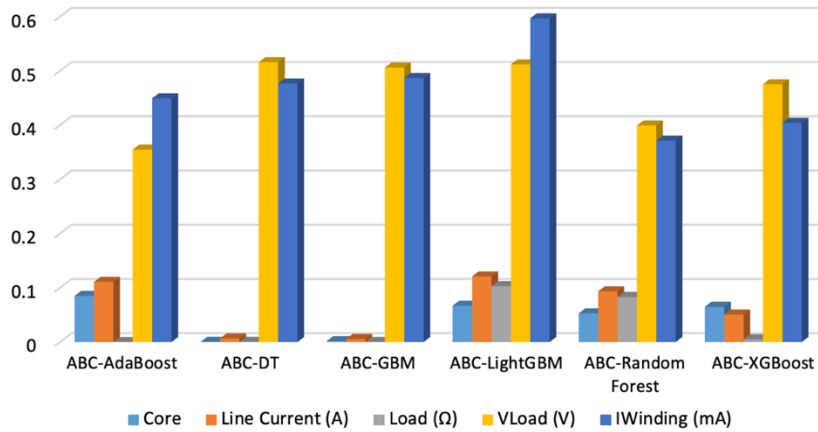


Figure 10: Feature importance of decision tree-based models

Although the dataset covers a variety of core types and current ranges, all experiments were conducted under controlled laboratory conditions. As a result, the models may be subject to bias or overfitting to the specific experimental configurations used. This limitation should be considered when interpreting the predictive performance. To mitigate this issue, future work could explore data augmentation techniques, cross-validation with independently acquired datasets, or real-world testing under varying environmental and operational conditions. These steps would help enhance the generalization capability and trustworthiness of the machine learning models in broader applications.

5 Conclusions

In this study, the harvested power of toroidal-type electromagnetic energy harvesters was accurately predicted using various machine learning regression models, including Lasso, AdaBoost, KNN, DT, GBM, LightGBM, Random Forest, and XGBoost. The models were trained and evaluated using a comprehensive dataset specifically created for this study, named the Toroidal-Energy-Harvesting Dataset. To further enhance model performance, these models were optimized using the ABC optimization technique. Among the evaluated models, the ABC-XGBoost demonstrated the highest predictive performance, achieving superior accuracy and robustness in estimating the harvested power of the energy harvesters.

Despite the promising results, this study has certain limitations. The dataset used for model training and evaluation may contain biases, which could limit the generalizability of the models to other types of electromagnetic energy harvesters. Addressing this limitation, future work could broaden the experimental scope by incorporating additional core shapes, a wider selection of magnetic materials, various frequency ranges, and environmental influences such as temperature and humidity. Such extensions are expected to contribute to the development of more generalizable and robust models that better reflect real-world deployment scenarios of electromagnetic energy harvesting systems. Additionally, exploring advanced feature

engineering techniques such as domain-specific feature extraction, and automated feature construction through deep learning could further enhance model accuracy and reliability. As a further enhancement, future work will focus on exploring model compression and lightweight architectures to enable real-time power prediction on low-power edge devices, which are commonly used in energy-harvesting applications. This approach aims to improve the practical applicability of the proposed model by reducing computational complexity without compromising prediction accuracy. Techniques such as pruning, quantization, and knowledge distillation will be considered to optimize the model for deployment on resource-constrained devices. Moreover, additional experiments will be conducted to validate the optimized models under real-world conditions, ensuring their robustness and adaptability to diverse operating environments. In summary, this study highlights the effectiveness of machine learning techniques, particularly when integrated with ABC optimization, in accurately predicting the output power of toroidal-type electromagnetic energy harvesters.

Data Availability

The data underpinning the analysis reported in this paper are deposited on a GitHub repository located at <https://github.com/kadirileri/Toroidal-Energy-Harvesting>.

Acknowledgements

This work was supported by the Bandırma Onyedü Eylül University Scientific Research Projects Coordination Unit under Project BAP-22-1004-009.

References

- [Balcı and Dalcalı, 2023] Balcı, M. S., and Dalcalı, A. (2023). Power Analysis of Toroidal Core Electromagnetic Energy Harvesters for Transmission Lines. *Engineering Technologies and Systems*, 4: 599-611.
- [Balcı and Dalcalı, 2024] Balcı, M. S., and Dalcalı, A. (2024). Design and application of optimum toroidal shaped electromagnetic energy harvesters for unmanned aerial vehicles. *International Journal of Numerical Modelling: Electronic Networks, Devices and Fields*, 37(3): e3260.
- [Bhosale et al., 2023] Bhosale, P. V., and Agashe, S. D. (2023). Investigation of piezoelectric energy harvester performance under frequency response and parameter prediction using hybrid DBN-SOA technique. *Energy Systems*, 1-23.
- [Breiman, 2001] Breiman, L. (2001). Random Forests. *Machine Learning*, 45: 5-32.
- [Cammarano et al., 2016] Cammarano, A., Petrioli, C., and Spenza, D. (2016). Online Energy Harvesting Prediction in Environmentally Powered Wireless Sensor Networks. *IEEE Sensors Journal*, 16: 6793-6804.
- [Chen and Guestrin, 2016] Chen, T., and Guestrin, C. (2016). Xgboost: A scalable tree boosting system. In *Proceedings of the 22nd acm sigkdd international conference on knowledge discovery and data mining*, pp. 785-794.
- [Ferreira, 2022] Ferreira, A. J. (2022). Boosting the Prediction of Extreme Values.

- [Freund and Schapire, 1996] Freund, Y., and Schapire, R. E. (1996). Experiments with a new boosting algorithm. In *13th International Conference on International Conference on Machine Learning*, 148-156.
- [Friedman, 2001] Friedman, J. H. (2001). Greedy function approximation: a gradient boosting machine. *Annals of statistics*, 1189-1232.
- [Gao et al., 2023] Gao, M., Herrera, H. L., and Moon, J. (2023) Optimization of Core Size and Harvested Power for Magnetic Energy Harvesters based on Cascaded Magnetics. In *IEEE Applied Power Electronics Conference and Exposition*, 2926-2932.
- [Gao et al., 2024] Gao, M., Yi, L., and Moon, J. (2024), Enabling Multiple Harvesting Windows in Magnetic Energy Harvesting via Reverse Flux Desaturation. *IEEE Journal of Emerging and Selected Topics in Power Electronics*, 12: 2517-253.
- [Hoang and Ebeid, 2023] Hoang, V. D., and Ebeid, E. (2023). Advanced Magnetic Energy Harvester for Charging Drones from Overhead Powerlines. In *25th European Conference on Power Electronics and Applications*, 1-9.
- [Huang et al., 2021] Huang, H., Li, C., Ye, K., Liu, H., Yang, H., Du, X. (2021). An Accurate Model of Magnetic Energy Harvester for Harvesting Maximum Power Based on Fitting Method. In *2nd China International Youth Conference on Electrical Engineering*, 1-6.
- [Ileberi et al., 2021] Ileberi, E., Sun, Y., and Wang, Z. (2021). Performance Evaluation of Machine Learning Methods for Credit Card Fraud Detection Using SMOTE and AdaBoost. *IEEE Access*, 9: 165286–165294.
- [Karaboğa and Basturk, 2007] Karaboga, D., and Basturk, B. (2007). A powerful and efficient algorithm for numerical function optimization: Artificial bee colony (ABC) algorithm. *Journal of Global Optimization*, 39: 459–471.
- [Ke et al., 2017] Ke, G., Meng, Q., Finley, T., Wang, T., Chen, W., Ma, W., Ye, Q., and Liu, T. Y. (2017). Lightgbm: A highly efficient gradient boosting decision tree. *Advances in neural information processing systems*, 30.
- [Kumar et al., 2022] Kumar, S., Kolekar, T., Kotecha, K., Patil, S., and Bongale, A. (2022). Performance evaluation for tool wear prediction based on Bi-directional, Encoder–Decoder and Hybrid Long Short-Term Memory models. *International Journal of Quality and Reliability Management*, 39: 1551–1576.
- [Kuo et al., 2018] Kuo, R. J., Huang, S. B. L., Zulvia, F. E., and Liao, T. W. (2018). Artificial bee colony-based support vector machines with feature selection and parameter optimization for rule extraction. *Knowledge and Information Systems*, 55: 253–274.
- [Kushwah et al., 2022] Kushwah, J. S., Kumar, A., Patel, S., Soni, R., Gawande, A., and Gupta, S. (2022). Comparative study of regressor and classifier with decision tree using modern tools. *Materials Today: Proceedings*, 56(6): 3571–3576.
- [Li and Han, 2022] Li, L., and Han, C. (2022). ASARIMA: An Adaptive Harvested Power Prediction Model for Solar Energy Harvesting Sensor Networks. *Electronics*, 11: 2934.
- [Lin et al., 2021] Lin, P., Hu, G., Li, C., Li, L., Xiao, Y., Tse, K. T., and Kwok, K. C. S. (2021). Machine learning-based prediction of crosswind vibrations of rectangular cylinders. *Journal of Wind Engineering and Industrial Aerodynamics*, 211: 104549.
- [Liu and Zhou, 2024] Liu, S., and Zhou, D. J. (2024). Using cross-validation methods to select time series models: Promises and pitfalls. *British Journal of Mathematical and Statistical Psychology*, 77 (2): 337–355.

- [Maruf et al., 2016] Maruf, S., Javed, K., and Babri, H. A. (2016). Improving Text Classification Performance with Random Forests-Based Feature Selection. *Arabian Journal for Science and Engineering*, 41: 951–964.
- [Monagle et al., 2023] Monagle, D., Ponce, E., and Leeb, S. B. (2023). Resonant Circuits for Split-Core Magnetic Energy Harvesters. *IEEE Transactions on Industrial Electronics*, 71(8): 9932 – 9941.
- [Mouapi et al., 2020] Mouapi, A., Hakem, N., and Kandil, N. (2020). Piezoelectric Energy Harvesting Prediction and Efficient Management for Industrial Wireless Sensor. *Applied Sciences*, 10 (23): 8486.
- [Ng and Newton, 2022] Ng, T. L., and Newton, M. A. (2022). Random weighting in LASSO regression. *Electronic Journal of Statistics*, 16: 3430–3481.
- [Pekel, 2020] Pekel, E., (2020). Estimation of soil moisture using decision tree regression. *Theoretical and Applied Climatology*, 139: 1111–1119 .
- [Qian et al., 2020] Qian, J., Tanigawa, Y., Du, W., Aguirre, M., Chang, C., Tibshirani, R., Rivas, M. A., and Hastie, T. (2020). A fast and scalable framework for large-scale and ultrahigh-dimensional sparse regression with application to the UK Biobank. *PLoS Genetics*, 16 (10):e1009141.
- [Riba et al., 2023] Riba, J. R., Arbat, R., Ndong, Y. O., and Eguilaz, M. M. (2023). Exploring the Limitations of Electric Field Energy Harvesting. *Electronics*, 12: 3626.
- [Sabat et al., 2023] Sabat, W., Klepacki, D., Kamuda, K., Kurylo, K., and Mihulowicz, P. J. (2023). Efficiency Measurements of Energy Harvesting from Electromagnetic Environment for Selected Harvester Systems. *Electronics*, 12: 4247.
- [Song et al., 2023] Song, T., Ding, L., Yang, L., Ran, J., and Zhang, L. (2023). Comparison of machine learning models for performance evaluation of wind-induced vibration piezoelectric energy harvester with fin-shaped attachments. *Ocean Engineering*, 280: 114630.
- [Suzuki et al., 2016] Suzuki, C., Itoh, T., Umezū, K., and Motohashi, Y. (2016). A scatterplot-based visualization tool for regression analysis. In *International Conference on Information Visualisation, Institute of Electrical and Electronics Engineers Inc.*, 75–80.
- [Toroidal-Energy-Harvesting, 2025] 2025, Toroidal-Energy-Harvesting Dataset is available at <https://github.com/kadirileri/Toroidal-Energy-Harvesting>.
- [Uddin et al., 2022] Uddin, S., Haque, I., Lu, H., Moni, M. A., and Gide, E. (2022). Comparative performance analysis of K-nearest neighbour (KNN) algorithm and its different variants for disease prediction. *Scientific Reports*, 12: 6556.
- [Veney and D’Souza, 2023] Veney, J., and D’Souza, K. (2023). Frequency tunable electromagnetic vibration energy harvester using piecewise linear nonlinearity. *Proceeding of the Royal Society A*, 479: 20230207.
- [Wang et al., 2024] Wang, S., Chen, Y., Cui, Z., Lin, L., and Zong, Y. (2024). Diabetes Risk Analysis based on Machine Learning LASSO Regression Model. *Journal of Theory and Practice of Engineering Science*, 4(1): 58–64.
- [Xiong and Yao, 2021] Xiong, L., and Yao, Y. (2021). Study on an adaptive thermal comfort model with K-nearest-neighbors (KNN) algorithm. *Building and Environment*, 202: 108026.
- [Ye et al., 2024] Ye, K., Liu, Z., Liu, C., Yang, A., Yuan, H., Rong, M., and Wang, X. (2024). A Novel Method of Maximum Power Point Reaching for Magnetic Field Energy Harvesting Based

on a Low-Power Analog Control Circuit. *IEEE Transactions on Power Electronics*, 39 (1): 1888-1897.

[Yin et al., 2024] Yin, P., Han, H., Tang, L., Tan, X., Guo, M., Xia, C., and Aw, K. C. (2024). Kresling origami-inspired electromagnetic energy harvester with reversible nonlinearity. *Smart Materials and Structures*, 33(3): 035043

[Yuan et al., 2023] Yuan, Z., Ge, Y., Wei, J., Yuan, S., Liu, R., and Mo, X. (2023). Energy Prediction for Energy-Harvesting Wireless Sensor: A Systematic Mapping Study. *Electronics*, 12: 4304

[Zhang et al., 2021] Zhang, C., Hu, G., Yurchenko, D., Lin, P., Gu, S., Song, D., Peng, H., and Wang, J. (2021). Machine learning based prediction of piezoelectric energy harvesting from wake galloping. *Mechanical Systems and Signal Processing*, 160: 107876.