Characterization of the Quality Factor in Spiral Coil Designs for High-Frequency Wireless Power Transfer Systems using Machine Learning

Minki Kim, Minoh Jeong, Martina Cardone and Jungwon Choi
Department of Electrical and Computer Engineering
University of Minnesota, Twin Cities
Minneapolis, USA
kim00756@umn.edu

2022 IEEE 23rd Workshop on Control and Modeling for Power Electronics (COMPEL) | 978-1-6654-1081-6/22/$31.00 ©2022 IEEE | DOI: 10.1109/COMPEL53829.2022.9830005

Abstract—**This paper presents a machine learning-based characterization of the quality (Q) factor in spiral coil designs for wireless power transfer systems operating at MHz frequencies. Due to skin and proximity effects, at such frequencies, it is challenging to estimate the Q factor of the coupling coils, which is a critical parameter to determine the system's efficiency. A three-dimensional (3D) electromagnetic (EM) simulator allows us to precisely analyze the performance of different coil structures. However, the long processing time in the simulator is a bottleneck for quickly optimizing the coil design. To overcome this issue, we here propose a design method with a feed-forward neural network (FNN) to predict the parameters of the spiral coil. The FNN leverages the data set collected via the 3D quasi-static EM field simulator to train a predictor using the stochastic gradient descent algorithm. After optimization, the FNN model estimates the Q factor of the spiral coil without any delay. The proposed algorithm shows an accuracy larger than 96% under an arbitrary structure. Moreover, the proposed coil design method significantly reduces the computation time and hence, the analysis complexity.**

Index Terms—**Spiral Coil Design, High-Frequency Wireless Power Transfer, Machine Learning.**

I. Introduction

In wireless power transfer (WPT) systems, the design of the coupling coils is critical for achieving a high conversion efficiency. High operating frequencies, such as 6.78 MHz and 13.56 MHz, are generally beneficial for obtaining amenable (e.g., small and light) WPT systems with long transmission distances. To estimate the efficiency of a WPT system, the coils are modeled by the lumped elements, e.g., the resistance R, inductance L and capacitance C, which determine the quality Q factor of the coils, that also depends on the operating frequency [1], [2]. During the operation of a WPT system, the electromagnetic (EM) fields in a coil to transfer energy usually cause power losses, such as radiation and conduction losses. Radiation losses are typically negligible for a small size coil due to the wavelength in the 10's of MHz operation, and conduction losses are mainly dependent on the value of R of the coil [1]. Sadly, at MHz frequencies, because of the proximity, skin effect, and other concealed losses, it is very challenging to obtain mathematical expressions for R. Moreover, the parasitic parallel capacitance is no longer negligible in the coil analysis at high frequencies [3], [4]. The stray capacitance, in fact, resonates with L and forms a parasitic parallel self-resonance frequency. This affects the performance of the WPT system when the operating frequency and the parasitic resonance frequency are close to each other [5]. In the case of a series resonant-based compensation network (e.g. Series-Series, Series-Parallel and LCC compensation network) of the WPT system, the parasitic parallel capacitance causes an unintentional degradation under the coupling coefficient or load variance conditions [6], [7]. Thus, it is necessary to consider the coil parasitic capacitance and minimize its effect. The stray capacitance of a spiral coil depends on the diameter of each turn, the number of turns, the pitch, and the conductor permittivity. However, the self-capacitance is difficult to extract accurately due to the nonlinear adjacent winding capacitance, which depends on the structure and shape of the coil [3]–[5]. There are various software simulators, based on numerical analyses, for the coil design optimization. Three-dimensional (3D) finite element method (FEM) solvers, such as Ansys-HFSS and COMSOL, are the most representative simulators for high-frequency electronic components. Also, a 3D quasi-static electromagnetic field simulator, such as Ansys-Q3D, can be used to extract directly the R, L, and C values [8], [9]. However, such conventional methods consume a significant amount of time to calculate the parameters and find the optimized coil design.

In this paper, we propose a new method to estimate the Q factor for spiral coils in a WPT system using machine learning (ML). Our ML method leverages data collected using the Ansys-Q3D simulator to train a model aimed at efficiently predicting the Q factor, i.e., the figure-of-merit (FOM) of the coil performance. The proposed method shows high accuracy ($> 91\%$) when we have enough data (> 2000 cases). These results show the feasibility of our proposed method for real-world applications that have enough data or coils' samples.

II. Proposed Spiral Coil Design

A. Spiral Coil for High-Frequency WPT Systems

The structure of a spiral coil consists of the outer diameter (D_o), inner diameter (D_i), number of turns (N), pitch size (p),

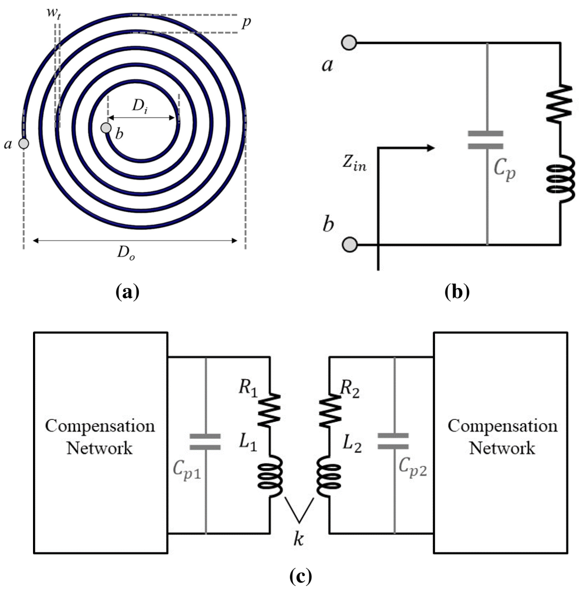


Fig. 1: Structure of a spiral coil: (a) design parameters; (b) lumped elements model; and (c) WPT system structure.

and wire thickness (w_t) of the coil, as illustrated in Fig. 1a. The equivalent circuit model of the spiral coil is shown in Fig. 1b. In particular, the input impedance (Z_{in}) is computed as,

$$\begin{aligned} Z_{in} = R_{eq} + jwL_{eq} &= \frac{1}{jwC_p}||(R + jwL) \\ &= \frac{R}{1 - 2C_pLw^2 + R^2C_p^2w^2 + L^2C_p^2w^4} \\ &\quad + j\frac{wL - wC_pR^2 - C_pL^2w^3}{1 - 2C_pLw^2 + R^2C_p^2w^2 + L^2C_p^2w^4}, \end{aligned} \tag{1}$$

where the coil resistance is R, the inductance is L, the parasitic capacitance[1] is C_p, and the angular frequency is w. The resistance R encompasses all the losses in the coil and its value is complex to estimate accurately at MHz frequencies because of the DC resistance, proximity, and skin-depth effects. To calculate the efficiency of a WPT system, we find the Q factor by using the input impedance, as shown below,

$$Q = \frac{X_{eq}}{R_{eq}} = \frac{w\left(L - C_pR^2 - C_pL^2w^2\right)}{R}, \tag{2}$$

where R_{eq} is the equivalent resistance, and X_{eq} is the equivalent reactance.

Then, the calculated Q value for the coupling coils are used to find the efficiency of the WPT system. Fig. 1c shows a general two-coil WPT system when the designed coils are used for the transmitter/receiver. The maximum conversion efficiency of the WPT system is denoted as η_{max} and expressed as,

$$\eta_{max} = \frac{k^2Q_1Q_2}{\left(1 + \sqrt{1 + k^2Q_1Q_2}\right)^2}, \tag{3}$$

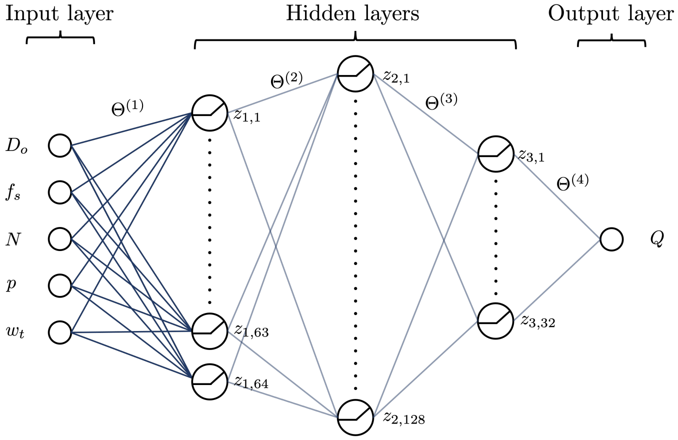


Fig. 2: FNN with three hidden layers and ReLU activation function.

where Q_1, Q_2 are the quality factors of L_1, L_2, respectively, and k is the coupling coefficient between the two coils [10].

Extracting the value of the Q factor using conventional EM field simulators requires a considerable amount of time. For instance, it takes approximately $10 \sim 15$ hours for 50 different designs ($12 \sim 18$ minutes/design) using Ansys-HFSS, and about $2 \sim 4$ hours for 50 designs ($2 \sim 5$ minutes/design) using Ansys-Q3D. Moreover, the simulation time depends on the structure, solution setup, and computing resources. It is clear that such long simulation times represent a bottleneck to quickly optimize the coil design. Motivated by this observation, we next propose a ML-based spiral coil design, as we describe in what follows.

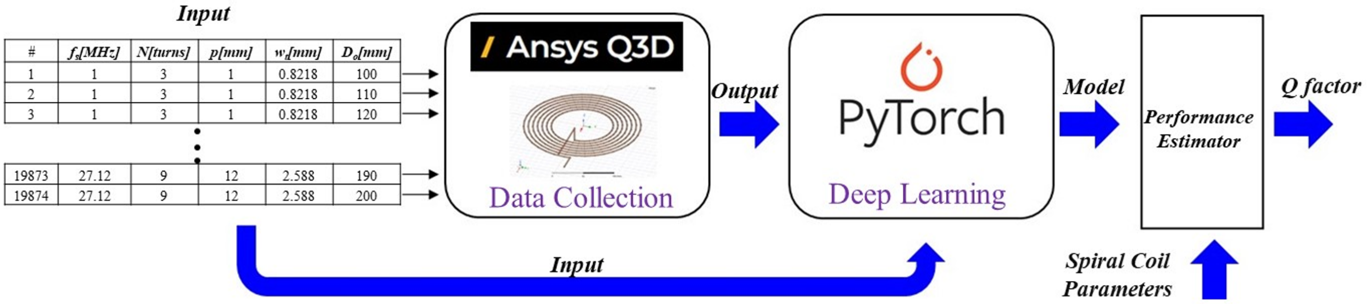
B. Prediction using Deep Learning

ML [11] models are promising methods to accurately approximate a large class of functions between input features and outputs, as long as enough data samples, that appropriately represent the end-to-end relationship, are available. Moreover, deep neural networks (DNN) [12] models yield unprecedented performance for various classification and regression tasks. A basic DNN model consists of the input layer, output layer, and several hidden layers. Each layer has a number of neurons that first perform a summation on their inputs, which is then followed by a non-linear activation function. By leveraging the non-linearity of the activation function, DNN can suitably approximate almost any function.

Estimating the Q factor in (2), which is a continuous real value, is cast as a regression problem. Thus, a properly designed DNN is leveraged to learn the true function between (D_o, f_s, N, p, w_t) and Q based on the available dataset[2]. Towards this end, we employ a feed-forward neural network (FNN), which consists of fully connected layers, i.e., all neurons in a layer are connected to every neuron in the next layer. Fig. 2 represents a simple FNN model consisting of five input features, one output, and three hidden layers.

[1] As highlighted in Section I, at MHz frequencies, the parasitic capacitor must be considered because it induces a parasitic self-resonance in the coil.

[2] Here, f_s denotes the frequency, and the other parameters are defined in Section II-A.



#	f_s[MHz]	N[turns]	p[mm]	w_t[mm]	D_o[mm]
1	1	3	1	0.8218	100
2	1	3	1	0.8218	110
3	1	3	1	0.8218	120
			⋮		
19873	27.12	9	12	2.588	190
19874	27.12	9	12	2.588	200

Fig. 3: System architecture including the data collection, FNN model training and performance evaluation for the spiral coil design.

In particular, each neuron in the hidden and output layers first computes a summation over all of the incoming values. Then, the neuron applies a non-linear activation function to the computed value and outputs the result of the activation function. Whenever there exists a next layer, the output of a neuron is multiplied by a corresponding model parameter and is conveyed to every neuron in the next layer.

With reference to Fig. 2, we denote by $z_{\ell,i}$ the output of the i-th neuron in the ℓ-th layer, and the operation of the neuron is formally expressed as

$$z_{\ell,i} = \sigma\left(\left\langle \Theta_i^{(\ell)}, u_\ell \right\rangle + b_i^{(\ell)}\right), \qquad (4)$$

where: (i) $\sigma(\cdot)$ is a non-linear activation function; (ii) $\langle \cdot, \cdot \rangle$ denotes the inner product; (iii) $\Theta_i^{(\ell)}$ is the weight vector for the i-th neuron in the ℓ-th hidden layer, and $\Theta^{(\ell)}$ is obtained by stacking together all the $\Theta_i^{(\ell)}$'s; (iv) u_ℓ is the input vector of the ℓ-th layer; and (v) $b_i^{(\ell)}$ is the bias term. We set the $z_{0,i}$'s (i.e., the input parameters) to be the parameters of the problem of interest, i.e., D_o, f_s, N, p, and w_t. We design a FNN model that consists of three hidden layers (see Fig. 2), where each layer has $64, 128$, and 32 neurons, respectively, followed by the rectified linear unit (ReLU) activation function, that is $\mathrm{ReLU}(x) = \max\{0, x\}$. By calculating $z_{\ell,i}$ for all i's, the output vector (denoted by z_ℓ) of the ℓ-th layer is obtained by stacking together the $z_{\ell,i}$'s, and it can be used as the input vector of the $(\ell+1)$-th layer if $\ell \in \{1, 2, 3\}$ or as the output of the FNN if $\ell = 4$. The estimated output value, denoted as $\hat{Q}$, can then be expressed as $\hat{Q} = z_{4,1} = \sum_{k=1}^{32} \Theta_{1,k}^{(4)} z_{3,k} + b_1^{(4)}$, where $\Theta_{1,k}^{(4)}$ is the k-th entry of $\Theta_1^{(4)}$.

The number of neurons in each layer is an important hyper-parameter since it controls the accuracy of a function approximation. Roughly speaking, the more neurons a DNN model has, the better result the model can produce. This trend also occurs as a function of the number of hidden layers and data samples. However, having too many parameters in the model may lead to over-fitting. This is a case where the ML model strongly memorizes its training dataset and yields a poor generalized performance (i.e., test performance) if a new unseen data (e.g., test data) is different (in a distribution sense) from the training data. To mitigate the over-fitting phenomenon in our model, we divide our dataset into three different sets, namely training data, validation data, and test data. The training data is used to train the ML model, and the validation data is used to evaluate the model performance (i.e., validation loss) during the training phase. The prediction performance is evaluated based on the test data, which is unknown to the ML model. In particular, we select the model parameters at the point before which the validation loss increases.

The loss function we used to train the FNN model is the well-known mean squared error (MSE) defined as

$$L(f(\mathbf{X}), \mathbf{y}) = \frac{1}{N_D} \sum_{i=1}^{N_D} (f(\mathbf{x}_i; \Theta) - y_i)^2, \qquad (5)$$

where: (i) $f(\cdot; \Theta)$ is the FNN model with parameters Θ; (ii) $(\mathbf{X}, \mathbf{y})$ is the dataset; (iii) N_D denotes the number of data samples; and (iv) $(\mathbf{x}_i, y_i)$ is the i-th data sample. In our setting, we used $\mathbf{x}_i = (D_o, f_s, N, p, w_t)$ and $y_i = Q$ for the i-th data sample. During the training phase, the parameters Θ's were optimized by using the stochastic gradient descent algorithm with batch size equal to 160, and total number of epochs equal to 2000. Note that the selected number of epochs is smaller than the one leading to an over-fitting in our ML model. Once the FNN model is trained based on the given dataset $(\mathbf{X}, \mathbf{y})$, it readily produces $\hat{Q}$, i.e., the estimated value of the Q factor corresponding to the coil design parameters.

III. Spiral Coil Design using ML

Fig. 3 shows the overall block diagram for our spiral coil design using ML, including the data collection, the FNN model training, and the ML-based spiral coil performance evaluation. The Ansys-Q3D was used for the data collection for the spiral coil design. The five input parameters (D_o, f_s, N, p, and w_t) have a specific range of values, selected by uniformly distributing them with a specific resolution. The collected data was used to train the FNN within the PyTorch ML framework. Also, additional data was collected to verify the out-of-range estimation performance. The fully-trained FNN model was then used to estimate the Q factor of the spiral coil and compared to the collected data to check the accuracy.

A. Data Collection

The Ansys-Q3D simulator was used to extract the spiral coil characteristics for the training/evaluation data. The physical structure of the spiral coil is dictated by the geometrical parameters such as D_o, N, p, w_t and f_s, as shown in Fig. 4. The enameled magnet wire, which has a copper conductor covered by 80 μm polyester, was used for the winding of the spiral coil. The start and end wires were further extended for the lead connection.

The generated physical spiral coil was analyzed by the 3D quasi-static electromagnetic field simulator, which was used to extract the capacitance C, the AC resistance R_{AC} and the AC inductance L_{AC} of the designed spiral coil. TABLE I provides the adaptive setup for the design. The maximum number of passes represents the maximum number of mesh refinement cycles for the adaptive analysis. The adaptive analysis stops once the maximum number of passes has been completed. The minimum number of passes sets the minimum number of mesh refinement cycles. The Q3D simulator stops the analysis once this amount of passes has been finished. The minimum number of converged passes determines the amount of passes that must meet the convergence requirements before the adaptive analysis is terminated. The percent error sets the desired solution accuracy; smaller values produce more accurate results. Finally, the percent refinement per pass determines how many meshes are added at each iteration of the adaptive refinement process. We ran the simulations with general computing resources using a standard desktop, and each design required approximately $2 \sim 5$ minutes.

Data collection was divided into three sets, namely Dataset-I, Dataset-II, and Dataset-III. Dataset-I was mainly used to train the ML model, whereas Dataset-II and Dataset-III are an outside range of Dataset-I and were used to evaluate the performance. Each dataset is composed of five inputs, namely the design factors D_o, f_s, N, p, w_t. The output is the Q factor, which was computed from (2) using the values of R_{AC}, L_{AC}, and C extracted from Ansys-Q3D.

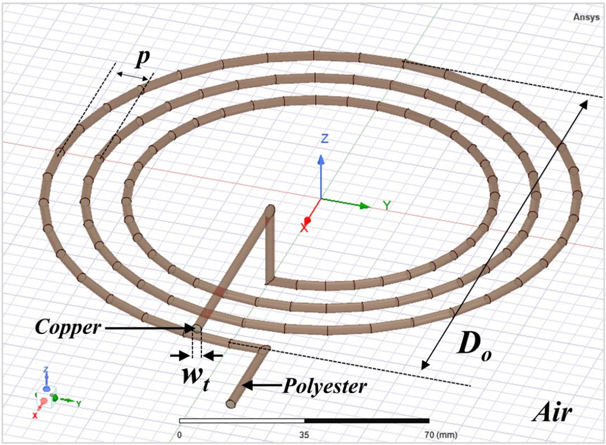


Fig. 4: Spiral coil design for data collection by using the Ansys-Q3D.

TABLE I: The analysis setup of the Ansys-Q3D for data collection.

Setting Parameters	Value
Solution Selection	C, R_{AC}, L_{AC}
Maximum # of Passes	10
Minimum # of Passes	1
Minimum Converged # of Passes	3
Percent Error	0.1%
Percent Refinement Per Pass	0.1%

TABLE II: The range of Dataset-I.

Variables	Values	# of Splits
f_s [MHz]	1, 6.78, 13.56, 20.34, 27.12	5
N [turns]	3, 4, 5, 6, 7, 8, 9	7
p [mm]	1, 2, 3, 4, 5, 6, 7, 8, 9, 10, 11, 12	12
w_t [mm]	0.822, 1.024, 1.290, 1.628, 2.052, 2.588	6
D_o [cm]	10, 11, 12, 13, 14, 15, 16, 17, 18, 19, 20	11
Total : $19,874$ $(27,720)$ cases		

TABLE III: The range of Dataset-II.

Variables	Values	# of Splits
f_s [MHz]	3, 10, 17, 23	4
N [turns]	3, 4, 5, 6, 7, 8, 9	7
p [mm]	1.4, 5.1, 7.9	3
w_t [mm]	0.9, 1.6, 2.3, 2.5	4
D_o [cm]	10.2, 13.9, 17.6, 19.1	4
Total : $1,091$ $(1,344)$ cases		

1) Dataset-I: This dataset has a large number of simulation cases and is used to train the FNN. The five input parameters $D_o, f_s, N, p,$ and w_t have a specific range of values, which were selected by uniformly distributing them with a specific resolution as shown in TABLE II. The operating frequency f_s was divided into 5 steps, from 1 MHz to 27.12 MHz. The number of turns N can take on 7 different values, from 3 to 9. The pitch p was divided into 12 steps, from 1 mm to 12 mm. The wire thickness w_t can take on 6 different values, from 20 AWG (= 0.822 mm) to 10 AWG (= 2.588 mm). The outer diameter D_o was separated into 11 values, from 100 mm to 200 mm. We extracted a total of $19,874$ samples obtained after removing some “impossible” cases from the collected $27,720$ samples. A total of $5,645$ “impossible” cases were removed since they had either too large p and N values or a too-small D_o, which led to a negative D_i,

$$D_i = D_o - 2N(p + w_t) > 0. \tag{6}$$

Then, $2,201$ “impossible” cases were filtered out because they had a negative Q value.

2) Dataset-II: As shown in TABLE III, this dataset is within the minimum/maximum range of Dataset-I, but does not overlap with Dataset-I. The operating frequency f_s is swept through 4 steps, namely [3 MHz, 10 MHz, 17 MHz, 23 MHz]. The turn N is swept through 7 steps, namely [3, 4, 5, 6, 7, 8, 9]. The pitch p is swept through 3 steps, namely [1.4 mm, 5.1 mm, 7.9 mm]. The wire thickness w_t is swept through 3 steps, namely [0.9 mm, 1.6 mm, 2.3 mm, 2.5 mm]. The outer diameter D_o is swept through 4 steps, namely [102 mm, 139 mm, 176 mm, 191 mm].

3) Dataset-III: As indicated in TABLE IV, this dataset includes the out of range values of Dataset-I. These values are computed as a $5\% - 15\%$ differences from the minimum and maximum ranges of Dataset-I. When one of the parameters is out-of-range, the other variables keep their values as in Dataset-I. The N is excluded from Dataset-III, because N has to be an integer and it is difficult to expand its range.

TABLE IV: The range of Dataset-III.

Variables	Values	# of cases
f_s [MHz]	0.85, 0.9, 0.95, 28.48, 29.83, 31.19	82
p [mm]	0.85, 0.9, 0.95, 12.6, 13.2, 13.8	72
w_t [mm]	0.691, 0.732, 0.772, 2.718, 2.847, 2.977	73
D_o [cm]	8.5, 9.0, 9.5, 21, 22, 23	82
	Total : 287 cases	

B. FNN Training

Our FNN model for the spiral coil design is trained using Dataset-I. As we described in Section II-B, our FNN model for estimating the Q factor consists of three hidden layers with $64, 128$ and 32 neurons in each layers, and the activation function is the ReLU function. The input values are the five key parameters D_o, f_s, N, p and w_t, and the output is the Q factor. Using Dataset-I, we optimized the model parameters Θ by making use of the stochastic gradient descent algorithm. In particular, to train the model, we utilized the ADAM optimizer with learning rate $\gamma = 0.001$. The loss function that we used in the training phase is the MSE as described in (5).

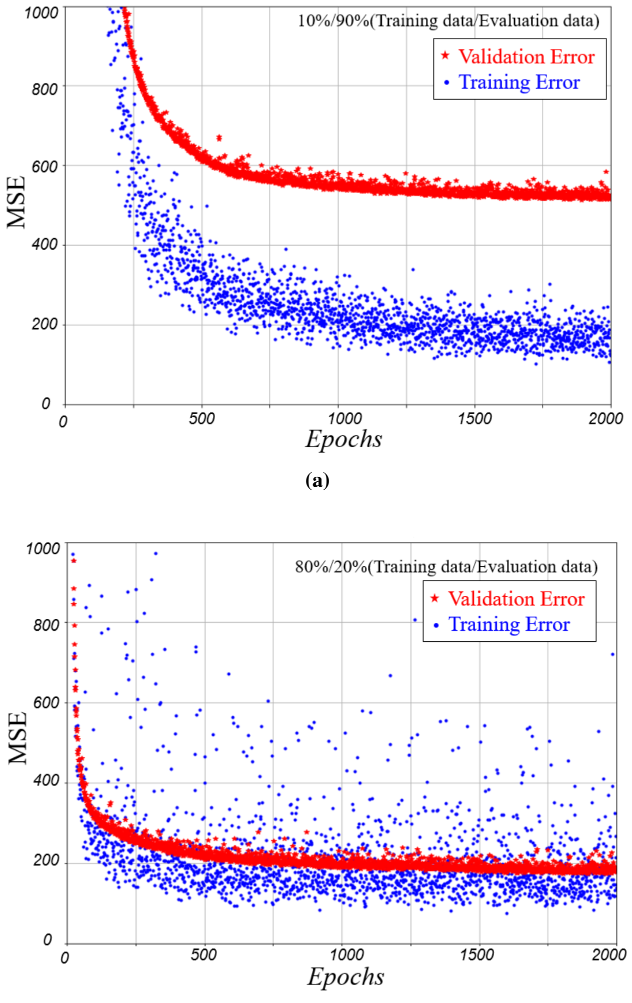
To guarantee a stable training, we normalized the input data. This ensures that the difference among the different data features (i.e., D_o, f_s, N, p and w_t) is not too large. Another significant benefit of such a normalization is a time reduction of the training phase. In particular, for an input feature $x \in \{D_o, f_s, N, p, w_t\}$, we used the standard score normalization defined as follows,

$$\overline{x} = \frac{x - \mu_x}{\sigma_x}, \tag{7}$$

where μ_x is the mean of x, and σ_x is the standard deviation of x. Since we collected all the $27,720$ data samples uniformly over the ranges described in Section III-A, for each data input x, we calculated μ_x and σ_x based on the values shown in Table II. For example, for the turn parameter N, we have $\mu_N = 6$ and $\sigma_N = 2$. These values, for $N = 5$, lead to $\overline{N} = -\frac{1}{2}$. The values of $\overline{x}$ are finally fed to our FNN.

Fig. 5 shows the training and validation losses as a function of the number of training epochs (i.e., a number that represents the amount of times that Dataset-I is used during training). As expected, the training and validation losses tend to decrease at each training epoch. During the training phase, we tried to have our FNN model avoid over-fitting. To this end, we chose the model parameters at a number of epochs after which the validation loss starts to increase. In particular, we observed that the over-fitting occurs in between $7,000$ and $10,000$ epochs. Moreover, from Fig. 5, we observe that the validation loss converges after $1,000$ epochs. Thus, to test our model, we chose the parameters when the number of epochs is $2,000$.

In order to observe the effect of the amount of data on the training, we randomly split Dataset-I ($19,874$ cases) into 8 different training datasets. In particular, the number of data samples in each training dataset gradually increases from 10% to 80% of the total number of samples in Dataset-I (the remaining data is used as validation data). For instance, the first training dataset contains 10% of the total samples in Dataset-I, the second training dataset contains 20% of the total samples in Dataset-I, and so on till the eighth training dataset that contains 80% of the total samples in Dataset-I. We also highlight that we selected the data samples in the training dataset number $i \in \{2, 3, \ldots, 8\}$ so as to include the data samples in the data set number $i - 1$.



(a)

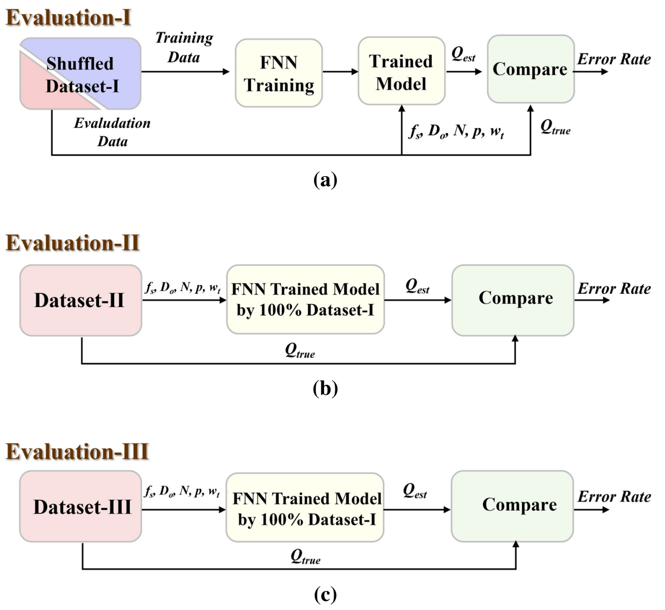
(b)

Fig. 5: FNN model training using Dataset-I: (a) 10%/90% ; (b) 80%/20%.

C. Evaluation of the ML-Based Spiral Coil Design

For the verification of the proposed spiral coil design, we assessed the performance of the ML-based design using Evaluation-I, Evaluation-II, and Evaluation-III, as illustrated in Fig. 6. All the three evaluations used the same FNN model trained by using Dataset-I, and the performance was measured using Dataset-I, Dataset-II, and Dataset-III, respectively.

1) Evaluation-I: This evaluation aims to validate our ML-based spiral coil design model in relation to the amount of

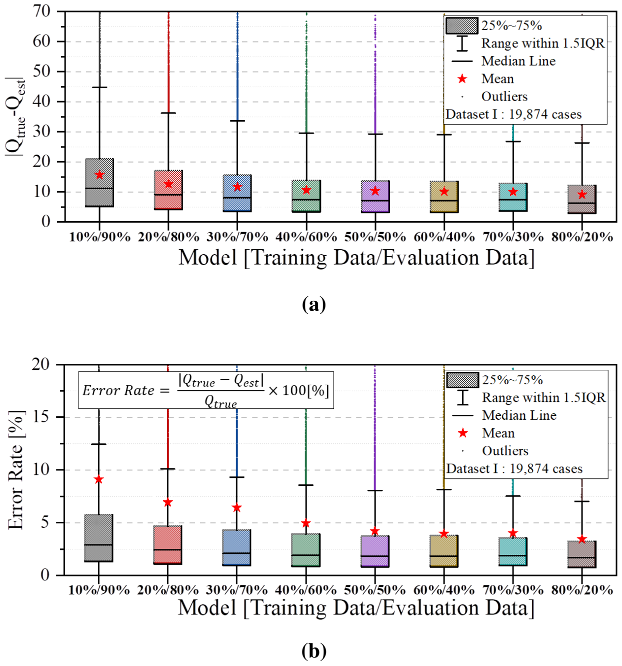


(a)

(b)

(c)

Fig. 6: Block diagrams of the evaluation procedures: (a) Evaluation-I ; (b) Evaluation-II ; (c) Evaluation-III.



(a)

(b)

Fig. 7: Performance of the training model using the Dataset-I: (a) error distribution ; (b) error rate distribution.

training data. Fig. 6a shows the block diagram of the evaluation procedure using Dataset-I. As mentioned in Section III-B, Dataset-I ($19,748$ cases) was randomly shuffled and separated into eight training datasets $D_i, i \in \{1, 2, \ldots, 8\}$, e.g., the first D_1 having 10% of the total samples in Dataset-I, the last D_8 having 80% of the total samples in Dataset-I. For each $i \in \{1, 2, \ldots, 8\}$, the data inside D_i was used to train the FNN-based model, and the remaining $(100\% - (10i)\%)$ data in Dataset-I was used to evaluate the performance. In what follows, for each D_i, we use the notation $(10i)\%/(100\% - (10i)\%)$ to denote that $(10i)\%$ of the data was used for training and the remaining $(100\% - (10i)\%)$ was used for validation.

The performance for the eight training models using Dataset-I is shown in Fig. 7. The eight models have the same number of training epochs and they only differ in the amount of training data. Thus, from Fig. 7, we can observe the prediction performance depending on the amount of available training data. From our experiment in Fig. 7, the training performance (measured in terms of the error and of the error rate), improves as the amount of training data increases. The average absolute error of the $10\%/90\%$ model is 15.6, while the one of the $80\%/20\%$ model is 9.1. Also, the average error rate of the $10\%/90\%$ model is 9.11% (90.89% accuracy), while the one of the $80\%/20\%$ model is 3.45% (96.55% accuracy). The result of the $10\%/90\%$ model empirically shows that our ML-based Q factor estimation performs well (the average error rate is below 10%) even if when less than $2,000$ training data samples are available.

2) Evaluation-II: This evaluation is illustrated in Fig. 6b and aims to validate our ML-based spiral coil design in scenarios that are not included in the training Dataset-I. In particular, we used all the samples inside Dataset-I to train the FNN, and then we evaluated the performance using Dataset-II (see TABLE III). As highlighted in Section III-A, even though Dataset-I and Dataset-II have the same range, they are disjoint. The error and the error rate distributions evaluated by using Dataset-II are shown in Fig. 8, and appear to follow a Gaussian distribution (red curve). Most of the error values $(Q_{\text{true}} - Q_{\text{est}})$ between the true Q factor and the estimated Q factor are concentrated around 0, and only a few error values are greater than 100. These can be considered as outliers and are due to an over-estimation of the Q factor. The histogram for the error rates is similar to the one for the error. The average absolute error for Evaluation-II is 23.41, and the average error rate is 6.37%. These results indicate that our ML-based estimation for the Q factor also performs well when the training and evaluation datasets have the same range, but are disjoint.

3) Evaluation-III: This evaluation is illustrated in Fig. 6c and it aims to verify the performance of our ML-based spiral coil design in scenarios where the evaluation dataset contains out-of-range values, i.e., values that are not in the range of those in the training Dataset-I. In particular, we trained our model using the entire Dataset-I, and then we evaluated the performance using Dataset-III (see TABLE IV). Fig. 9 shows the error rate incurred by each of the four out-of-range parameters in TABLE IV, while keeping the remaining variables within the range as in TABLE II. As shown in Fig. 9, f_s incurs the highest error rate, which is around 5.1%. This is because the number of unique values for f_s in the training Dataset-I is smaller than those of the other input variables. The result of Evaluation-III suggests that our ML-based model yields an estimated Q factor reasonably close to the true Q

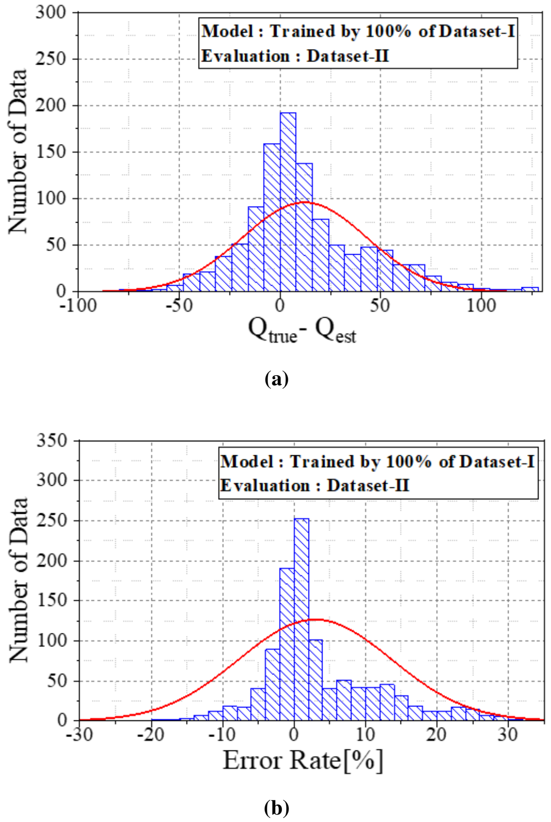


Fig. 8: Performance of the training model using Dataset-II: (a) error distribution ; (b) error rate distribution.

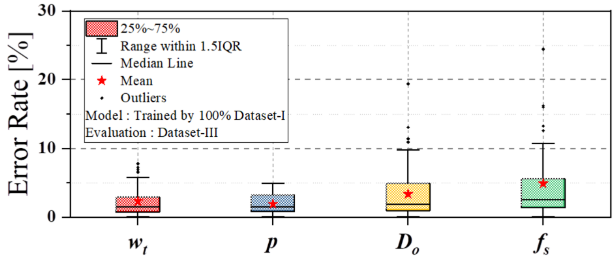


Fig. 9: Performance of the training model using Dataset-III.

factor even when the evaluation dataset contains values that are not in the range of the training dataset.

The Q values obtained using our proposed ML-based spiral coil design were compared to those obtained using the HFSS and the Q3D simulators, as shown in Fig. 10. Since we trained the model using data collected from the Ansys-Q3D simulator, our proposed model shows results closer to those of the Ansys-Q3D than those of the Ansys-HFSS simulator. This suggests that, if we have enough real-world data based on practical fabrication, our proposed ML-based coil design may output suitable results for real-world applications.

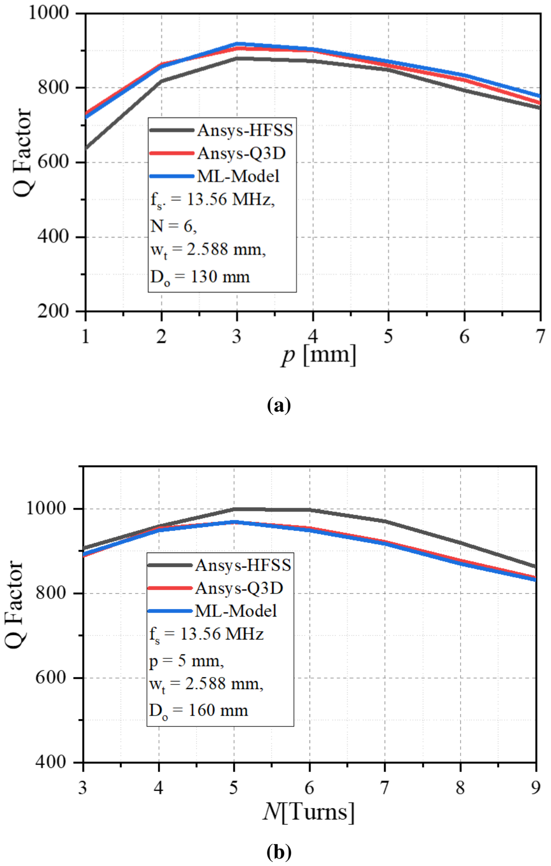


Fig. 10: Performance of the proposed spiral coil design: Comparison with Ansys-HFSS and Ansys-Q3D: (a) Pitch variance ; (b) Turns variance.

IV. Conclusion

This paper presented a characterization of the Q factor for spiral coil designs in a MHz-range WPT system using ML. The proposed method is simple and effective in estimating the performance of the spiral coil under the physical design environment. The proposed design method based on ML successfully predicts the Q factor by training over $19,874$ cases of the spiral coil characteristics.The Q factor of the proposed model has a high accuracy ($> 93\%$) compared to the actual value and it can be readily computed, hence significantly reducing the computation time incurred by conventional simulation based methods.

Acknowledgment

This material is based upon work supported by the National Science Foundation under Grant ECCS-2045239.

References

[1] B. H. Waters, B. J. Mahoney, G. Lee, and J. R. Smith, "Optimal coil size ratios for wireless power transfer applications," in *2014 IEEE International Symposium on Circuits and Systems (ISCAS)*, 2014, pp. 2045–2048.

[2] J. Cho, J. Sun, H. Kim, J. Fan, Y. Lu, and S. Pan, "Coil design for 100 khz and 6.78 mhz wpt system :litz and solid wires and winding methods," in *2017 IEEE International Symposium on Electromagnetic Compatibility Signal/Power Integrity (EMCSI)*, 2017, pp. 803–806.

[3] G. Grandi, M. Kazimierczuk, A. Massarini, and U. Reggiani, “Stray capacitances of single-layer air-core inductors for high-frequency applications,” in *IAS ’96. Conference Record of the 1996 IEEE Industry Applications Conference Thirty-First IAS Annual Meeting*, vol. 3, 1996, pp. 1384–1388 vol.3.

[4] Z. Jiang, P. Excell, and Z. Hejazi, “Calculation of distributed capacitances of spiral resonators,” *IEEE Transactions on Microwave Theory and Techniques*, vol. 45, no. 1, pp. 139–142, 1997.

[5] I. Lope, C. Carretero, and J. Acero, “First self-resonant frequency of power inductors based on approximated corrected stray capacitances,” *IET Power Electronics*, vol. 14, no. 2, pp. 257–267, 2021. [Online]. Available: https://ietresearch.onlinelibrary.wiley.com/doi/abs/10.1049/pel2.12030

[6] V. Shevchenko, O. Husev, R. Strzelecki, B. Pakhaliuk, N. Poliakov, and N. Strzelecka, “Compensation topologies in ipt systems: Standards, requirements, classification, analysis, comparison and application,” *IEEE Access*, vol. 7, pp. 120 559–120 580, 2019.

[7] C.-S. Wang, G. Covic, and O. Stielau, “Power transfer capability and bifurcation phenomena of loosely coupled inductive power transfer systems,” *IEEE Transactions on Industrial Electronics*, vol. 51, no. 1, pp. 148–157, 2004.

[8] M. Kim, S. Park, and H.-K. Jung, “An advanced numerical technique for a quasi-static electromagnetic field simulation based on the finite-difference time-domain method,” *Journal of Computational Physics*, vol. 373, pp. 917–923, 2018. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0021999118304935

[9] Q. Chen and A. Konrad, “A review of finite element open boundary techniques for static and quasi-static electromagnetic field problems,” *IEEE Transactions on Magnetics*, vol. 33, no. 1, pp. 663–676, 1997.

[10] S. Li and C. C. Mi, “Wireless power transfer for electric vehicle applications,” *IEEE Journal of Emerging and Selected Topics in Power Electronics*, vol. 3, no. 1, pp. 4–17, 2015.

[11] C. M. Bishop and N. M. Nasrabadi, *Pattern recognition and machine learning*. Springer, 2006, vol. 4, no. 4.

[12] I. Goodfellow, Y. Bengio, and A. Courville, *Deep learning*. MIT press, 2016.