Machine-Learned Models for Power Magnetic Material Characteristics

Paweł Leszczyński*, Kamil Kutorasiński†, Marcin Szewczyk*, *Senior Member, IEEE*, and Jarosław Pawłowski‡ *

Abstract—We present a general framework for modeling power magnetic materials characteristics using deep neural networks. Magnetic materials represented by multidimensional characteristics (that mimic measurements) are used to train the neural autoencoder model in an unsupervised manner. The encoder is trying to predict the material parameters of a theoretical model, which is then used in a decoder part. The decoder, using the predicted parameters, reconstructs the input characteristics. The neural model is trained to capture a synthetically generated set of characteristics that can cover a broad range of material behaviors, leading to a model that can generalize on the underlying physics rather than just optimize the model parameters for a single measurement. After setting up the model, we prove its usefulness in the complex problem of modeling magnetic materials in the frequency and current (out-of-linear range) domains simultaneously, for which we use measured characteristics obtained for frequency up to 10 MHz and H-field up to saturation.

Index Terms—power magnetics, materials modeling, deep neural networks, synthetic data, magnetic ring

I. INTRODUCTION

MAGNETIC components, such as inductors and transformers, are vital in power electronics devices requiring dedicated modeling efforts due to specific working conditions and underlying physical principles. No fully satisfactory first-principle models have yet been proposed for power magnetic materials, and currently efforts are underway to reproduce magnetic materials characteristics using machine learning frameworks [1]–[3]. After developing a theory that describes the underlying physics, one comes to the stage where it is necessary to choose the parameters of the model, most often adjusting them to obtain a correspondence between the model characteristics and the measurements. Depending on the model complexity, a typical approach is to optimize parameters by curve fitting using least-squares or one of the vast variety of optimization algorithms, such as gradient-based methods [4], simulated annealing [5], particle swarm optimizations [6], [7], or evolutionary algorithms [8], [9]. Regardless of the algorithm chosen, optimization is always the selection of a parameter for a given set of measurement data.

*Manuscript created Feb 16, 2024; *Division of Power Apparatus, Protection, and Control, Faculty of Electrical Engineering, Electrical Power Engineering Institute, Warsaw University of Technology, Koszykowa 75 St., 00-662 Warszawa, Poland †Department of Condensed Matter Physics, Faculty of Physics and Applied Computer Science, AGH University of Krakow, Reymonta 19 St., 30-059 Kraków, Poland ‡Institute of Theoretical Physics, Wrocław University of Science and Technology, Wyb. Wyspiańskiego 27 St., 50-370 Wrocław, Poland (*Corresponding author: Jarosław Pawłowski*).

With the advent of big data, deep learning, i.e. paradigm that uses deep neural networks (NNs), revolutionary breakthroughs in many conventional machine learning and pattern recognition tasks are a fact, such as computer vision or signal processing [10], [11]. The accompanied data-driven materials modeling approach [12]–[15], including data-driven magnetic materials modeling [1]–[3], also provides significant shift in materials science methodology from simple trial-and-error routine to intelligent discovery of new materials with demanding properties [16]–[20].

In this paper, we propose a similar approach but for learning *any* analytical model of a material in terms of intelligent fitting of its parameters. The only requirement for the model is that it must be differentiable to allow the fitting error to be passed back (backpropagated) during the NN training. In the following, we train a machine learning model composed of a deep NN to be able to select parameters of a given theoretical model over a wide range of material characteristics at the same time, not just to single measurement data, which from machine learning perspective would be just overfilling of this single characteristic. Our aim is to construct a model capable of generalizing across various features, not just one, which also implies that the neural network can capture the physics underlying the theoretical model.

Neural networks have been applied for modeling core losses or hysteresis loops of power magnetics (e.g., nanocrystalline rings) using simple few-layer perceptrons [21]–[27], or recurrent NN [28]. Also, recently introduced *MagNet* platform [29], [30] proves that NN can be utilized to effectively learn B-H loop of magnetic material via more advanced autoencoder architectures, similar to ours.

Nanocrystalline rings are currently used in many power and electronic devices [31] to meet the requirements imposed on the devices with respect to electromagnetic compatibility by suppressing electromagnetic interferences and insulation coordination by suppressing overvoltages occurring in these devices, in particular, overvoltages occurring in Gas-Insulated Switchgear [32]).

II. MODEL AND DATA GENERATION

.Various approaches to modeling magnetic materials in nonlinear conditions were proposed, including famous Jiles-Atherton (JA) model [33], [34], models based on Finite Element Method (FEM) that can describe local spatial dependence of magnetization [35], [36], or lumped element equivalent circuits (LEEC) [37], suitable for direct implementation in SPICE or EMTP simulators [38], [39].
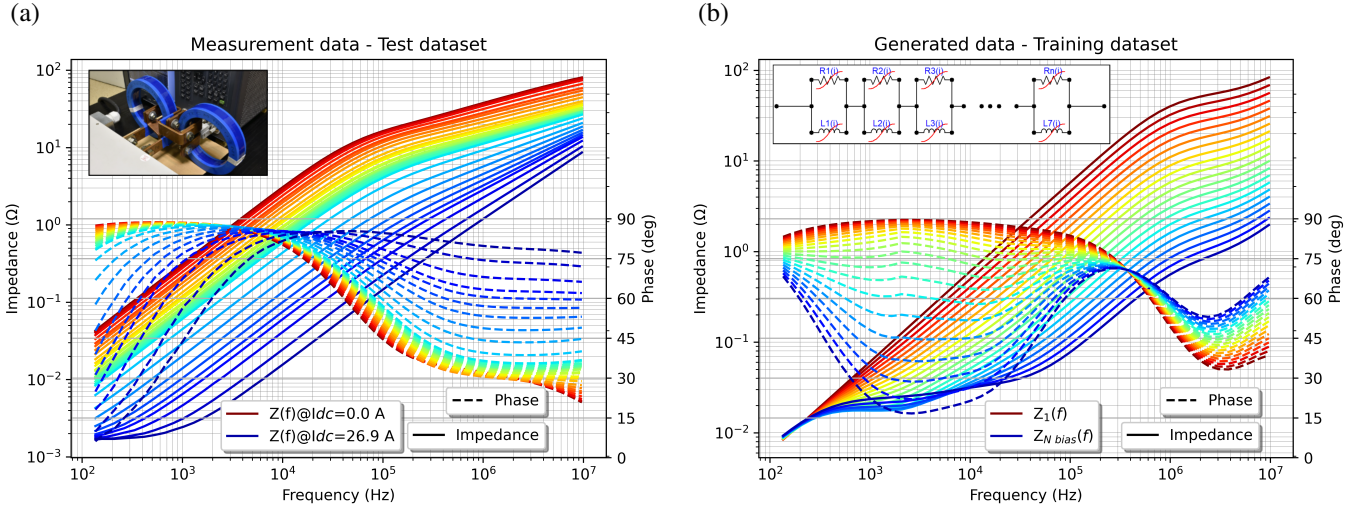
Figure 1. Impedance of a magnetic ring (a) measured as a function of frequency for various DC current biasings that controls operating point from ($I_{\text{DC}} = 0$ A, red curve) to the full saturation ($I_{\text{DC}} = 27$ A, blue curve), and (b) generated (random sample) using lumped element model with randomly selected outermost characteristics (red and blue one) and continuous transition between different curves. The generated data (b) will serve as a training set for the introduced neural network model, while the measurement ones (a) will be used to test the already trained NN model.

To show the capability of the proposed approach, we will focus on a phenomenological model of magnetic material in a wide frequency range capturing both linear and nonlinear H-field regimes. Here, we will use LEEC representing impedance characteristics of magnetic rings in both frequency and current domains, introduced in [40], and further analyzed in terms of its parameters optimization [41], also using AI methods [42].

To make our analysis realistic, we collect a set of measurements that will serve as a test set for the developed method. In Fig. 1(a), there is presented a family of impedances as a function of frequency f, taken for different DC-bias current I_{DC} which controls the ring magnetization: from a linear range (see red curves in Fig.1a) up to the ring saturation (blue curves in Fig.1a). The color of the measured impedance curves Z^{meas}, that changes gradually from red, for $I_{\text{DC}} = 0$ A, to blue, for $I_{\text{DC}} = 27$ A, ($N_{\text{bias}} = 31$ different values in total) shows impedances for different DC-bias current I_{DC} and frequency: $Z^{\text{meas}} = Z^{\text{meas}}(I_{\text{DC}}, f)$. Having families of characteristics, not just a single one, will be crucial when proving the model's capability to capture broad physics rather than just fitting a single feature. Methods for measuring magnetic rings over a wide range of frequencies and DC-bias current up to saturation are not a trivial task (especially reaching the saturation limit is challenging) and are described in detail in Ref. [43].

Let us now introduce the analytical model of a ring in the form of LEEC impedance synthesized out of the series of $M = 7$ L and R elements connected in parallel, as presented in Fig. 1(b), with the equivalent impedance in a form

$$Z^{\text{model}}(s) = \sum_{i=1}^{M} \frac{sR_i}{s + \frac{R_i}{L_i}}, \qquad (1)$$

where $s = j2\pi f$, and j is the imaginary unit. A series of parallel LR circuits was chosen as the best representation of the measurement data, whose impedance has a phase in the range of 0-90 degrees (indicating it is resistive-inductive). The number of $M = 7$ elements was selected to ensure generality
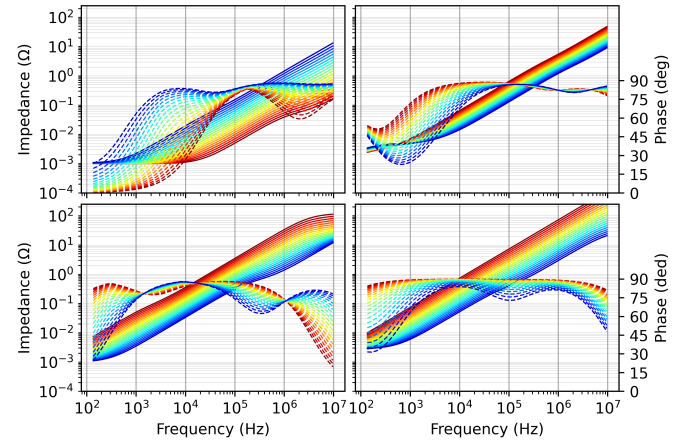

Figure 2. Four representative families of characteristics $Z_j(I_{\text{DC}}, f)$, where $j = 1 \ldots N_{\text{bias}}$, from the synthesized training dataset. Solid line denotes impedance amplitude and dashed line denotes impedance phase as in Fig.1b.

of the model and will be discussed in further sections. The goal for the NN will be to learn the model defined in Eq. 1 and predict its parameters, i.e., to find $\{L_i, R_i\}$ for a given frequency characteristics over the wide range of their possible behaviors. Therefore, to train the NN model and to force it to generalize over different behaviors, we have to synthesize the training dataset using the analytical model with random parameters. Generating data and training neural models on artificially synthesized datasets with controlled properties is currently a growing trend in machine learning [44].

The synthesized dataset will be composed of families of N_{bias} frequency characteristics gradually transforming into each other, which mimics the situation of continuous transition in the ring material operating regime; in our measurements controlled by the DC-bias current I_{DC}. Each pair of the outermost characteristics $Z_1(f)$ (for zero DC-Bias) and $Z_{N_{\text{bias}}}(f)$ (for full saturation) were generated from Eq. 1 for two

randomly selected sets of values (each of size $2M$): R_i in the range from 10^{-3} to 10^4 Ω, and L_i in the range from 10^{-7} to 10^{-4} H. Assumed ranges correspond to boundary values of these elements (equivalent resistance and inductance) during the measurements. Afterwards, for each pair of characteristics – example set is presented in Fig. 1(b) with outermost ones marked in red and blue – we calculate $N_{\text{bias}} - 2$ intermediate values at every frequency point with equal distances for real and imaginary parts separately. This way we get N_{bias} characteristics that form continuous transitions (across their full f range) from one outermost to another – colors between red and blue in Fig. 1(b), in accordance to measurements with different DC-bias currents. In total, we collected $N_{\text{train}} = 30\,000$ multidimensional families, each containing $N_{\text{bias}} = 21$ curves gradually transforming into each other. Examples of some other families of characteristics are presented in Fig. 2. These data will serve as the training set.

III. NEURAL NETWORK MODEL

Machine learning (ML) is a rapidly expanding field that has begun to revolutionize many areas, including magnetic design in power electronics [27]–[30]. ML is a branch of AI that focuses on developing algorithms and statistical models that enable computers to perform tasks without explicit instructions. Instead, they rely on patterns *learned* directly from data. An *autoencoder* (AE) is a type of NN used to learn efficient coding of data. It is trained in an unsupervised manner to ignore irrelevant "features" and learn to represent the data in a compressed form. AEs are composed of two parts: *encoder* processes the input data and compresses it into a smaller, dense representation, which captures the essence of the data, while *decoder*, working in reverse of the encoder, takes the compressed representation and reconstructs the original input as closely as possible.

After defining the generated training and measured test datasets, let us define the NN responsible for finding the model parameters. The network will take a form of AE as illustrated in Fig. 3, and trained in an unsupervised way. The role of the encoder is to predict the analytical model parameters $\{L_i, R_i\}$ (their values define the *latent space* of our AE), while the decoder will just calculate the frequency characteristic directly using Eq. 1 with parameters (latent space) found by the encoder.

The goal for the NN is to reconstruct input characteristics as accurately as possible, minimizing the reconstruction error, defined by the *loss* function. This means that in our training process, we don't require any true predictions (*labels*); only the frequency characteristics need to be reconstructed. This makes our training scheme fully unsupervised.

The standard solution would also be to use an NN as the decoder, but this approach makes less sense because we need the decoder only to train the encoder, which in turn, is essential for us (it predicts the analytical model parameters). Therefore, we successfully replaced the NN decoder with the analytical model itself (here defined by Eq. 1 for equivalent LEEC impedance) implemented as part of the NN (without trainable parameters, but using the model parameters ($\{L_i, R_i\}$)

predicted by the encoder) to be able to use the standard *backpropagation* of loss mechanism during the training.

Frequency characteristics are organized on a 1D evenly spaced grid along multiple decades on the log scale (50 points from 135 Hz to 10^7 Hz). To build an effective neural representation, we used convolutional neural network (CNN) layers as feature extractor parts, which is a common way to process data organized on regular grids [45], [46]. Then, we applied several fully connected (FC) layers serving as parameter predictors, which are based on previously found representations (by CNN). The detailed structure of the encoder network is presented in Fig. 3: three CNN feature extractor layers are followed by three FC layers with $2M$ (see. Eq.1) outputs in the last layer that predict the parameters. We select CNN as a feature extractor (backbone) part, which is common practice in signal or image processing [10], [11]; however, in principle, any other type of network could serve as backbone network at this stage, depending on applications – such as recurrent neural networks or nowadays very popular transformer networks [47] with a self-attention mechanism incorporated.

To train a NN model, we have to define a loss function $\mathscr{L}$, which in our case will be just the decoder (reconstruction) loss $\mathscr{L} = \mathscr{L}_{\text{dec}}$, i.e. the mean squared difference between input and output impedance, taken in logarithmic scale for real and imaginary part separately:

$$\mathscr{L}_{\text{dec}} = \text{MSLE}\left(\Re(Z), \Re(\hat{Z})\right) + \text{MSLE}\left(\Im(Z), \Im(\hat{Z})\right), \quad (2)$$

with $\hat{Z}$ meaning reconstructed characteristic (decoder output), and mean squared logarithmic error defined as:

$$\text{MSLE}\left(Z, \hat{Z}\right) = \frac{1}{N} \sum_{j=1}^{N} \left(\log(Z_j) - \log(\hat{Z}_j)\right)^2. \quad (3)$$

The AE is trained over the whole training set and the sum in Eq. 2 goes over $N = N_{\text{train}} \times N_{\text{bias}}$ frequency characteristics Z in the training set. However, in practice, during a single training step of the stochastic gradient descent (SGD) method, only a smaller random portion (mini-batch) is used, here $N_{\text{batch}} = 1024$. Training typically lasted 100-500 epochs up to the loss saturation. To validate the NN model performance after the training, we also define the relative error of the fitting (FE):

$$\text{FE} = \sqrt{\text{RE}\left(\Re(Z), \Re(\hat{Z})\right) + \text{RE}\left(\Im(Z), \Im(\hat{Z})\right)}, \quad (4)$$

with

$$\text{RE}\left(Z, \hat{Z}\right) = \frac{1}{N} \sum_{j=1}^{N} \left(\frac{\log(Z_j) - \log(\hat{Z}_j)}{\log(Z_j)}\right)^2. \quad (5)$$

IV. BASIC MODEL RESULTS

Let us now analyze the NN model performance. To verify the model's ability to generalize over the broad variety of characteristics we estimate the validation error using Eq. 4 but calculated after the training on some additional validation set generated in the same way as the training one. Table I proves that the model is capable of predicting parameters for a variety
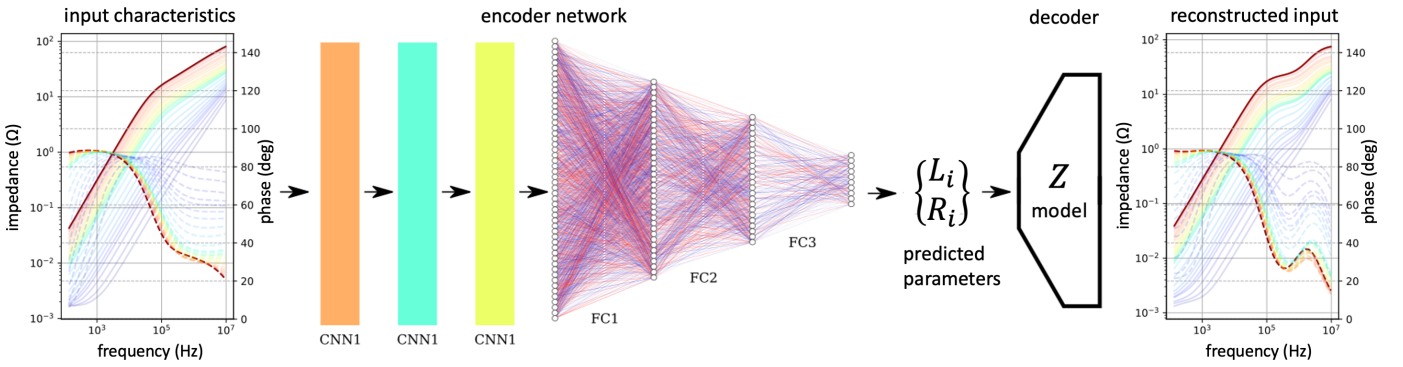
Figure 3. Neural network basic model with an autoencoder structure, which is of fundamental type and used in various fields of machine learning or image processing. The encoder is made of three convolutional layers (CNN) and three fully connected (FC) layers. The CNN layers are used as feature extractors, while the FC layers are used for parameters ($\{L_i, R_i\}$) predicting based on previously found representations. The decoder (Z) just implements an analytical formula for the impedance of the LEEC model. The goal of the network is to reconstruct the input characteristics at the output.

of characteristics (as example one presented in Fig. 1(b)), which is a crucial property, a necessary condition that must be met in order to have a chance of predicting parameters for unknown measured characteristic.

The result presented in 1st column of Table I shows that the basic version of the NN model (depicted in Fig. 3) generalizes quite well with the fitting error slightly above 8%.

Let us now perform tests and apply the model to measured data as presented in Fig. 1(a). Comparison of characteristics for the analytical model (defined in Eq. 1, with parameters predicted via the NN model) and the measured one, for selected I_{DC} values, are presented in Fig. 4(a,b) and shows quite good fitting performance over the broad range of DC-bias currents. It is worth recalling here that the measurement data was not a part of the training dataset. Moreover, we calculated the relative fitting error systematically, using Eq. 4 but now on the measured data, for every DC-bias current over the frequency range as presented in Fig. 4(c). In general, we observe that the error does not exceed 30%, which is quite a resonable result for fitting the model over so wide range of frequencies, impedance amplitudes, and DC-bias currents I_{DC}. On average it is even smaller and equals 12.54%.

V. SIAMESE NEURAL NETWORK MODEL

The results obtained for the basic NN model are acceptable but have one substantial drawback. If we look at Fig. 4(d) which presents values of characteristic frequencies defined as $\omega_i = R_i/L_i$ for each of LR pairs in a series (of size $M = 7$) for subsequent I_{DC} currents, we observe that the values are irregular and quite independent from each other which seems to be unphysical. In order to restore the more

realistic behavior which is more likely to be continuous (we do not expect to have any phase transitions in the model here), the standard NN model was extended to the Siamese model shown in Fig. 5. The idea behind the Siamese NNs, popular in many ML areas, such as face recognition [48], is to have two (or more) copies of a network that works in parallel and process different data samples at the same time. Two Siamese copies use (share) the same NN parameters (weights) while working in tandem (see Fig. 5) on two different input samples to compute two separate predictions. Throughout the training, the weights are effectively updated twice by applying distinct loss terms for each branch, along with an additional term that just compares the two predictions. Processing multiple samples at once enables to regularize of the model behavior with respect to the similarity between data samples by adding this extra term to the loss function. In our case we can process neighbor characteristics, i.e. for I_{DC} and $I_{\mathrm{DC}} + \Delta I$ – see Fig. 5, and enforce parameters predicted for that data points to change continuously, by including additional loss terms in a form:

$$l_{\mathrm{cont}}(L) = \frac{1}{N_{\mathrm{bias}}-1} \sum_{k=1}^{N_{\mathrm{bias}}-1} \log^2\left(\frac{L_{k+1}}{L_k}\right). \qquad (6)$$

The sum in Eq. 6 goes for every $N_{\mathrm{bias}} - 1$ neighboring pair in a given characteristics family. It penalizes too high a change of the parameters between neighbors. The total continuity loss should include M parameters and all N_{train} training families:

$$\mathscr{L}_{\mathrm{cont}}(L) = \frac{1}{MN_{\mathrm{train}}} \sum_{i=1}^{M} \sum_{j=1}^{N_{\mathrm{train}}} l_{\mathrm{cont}}(L_{i,j}). \qquad (7)$$

	basic NN	Siamese NN	modified Siamese NN	least squares [42]	differential evolution [42]	modified least squares [41]
model test error / validation error (for NN only)	12% / 8.3%	7.2% / 5.3%	5.2% / 4.5%	28%	8.4%	4.4%
model training or fitting time / evaluation time (for NN only)	0.5 h / ~1 s	1 h / ~1 s	1 h / ~1 s	~5 min	20-30 h	~15 h

Table I

RELATIVE FITTING ERROR FOR DIFFERENT MODIFICATIONS OF THE NN MODEL CALCULATED USING VALIDATION SYNTHETIC SET AND MEASURED TEST SET. FOR COMPARISON, TEST RESULTS FOR SIMILAR MATERIAL USING SOME OTHER METHODS FOLLOWING REFS. 42, 41 ARE ALSO PRESENTED.
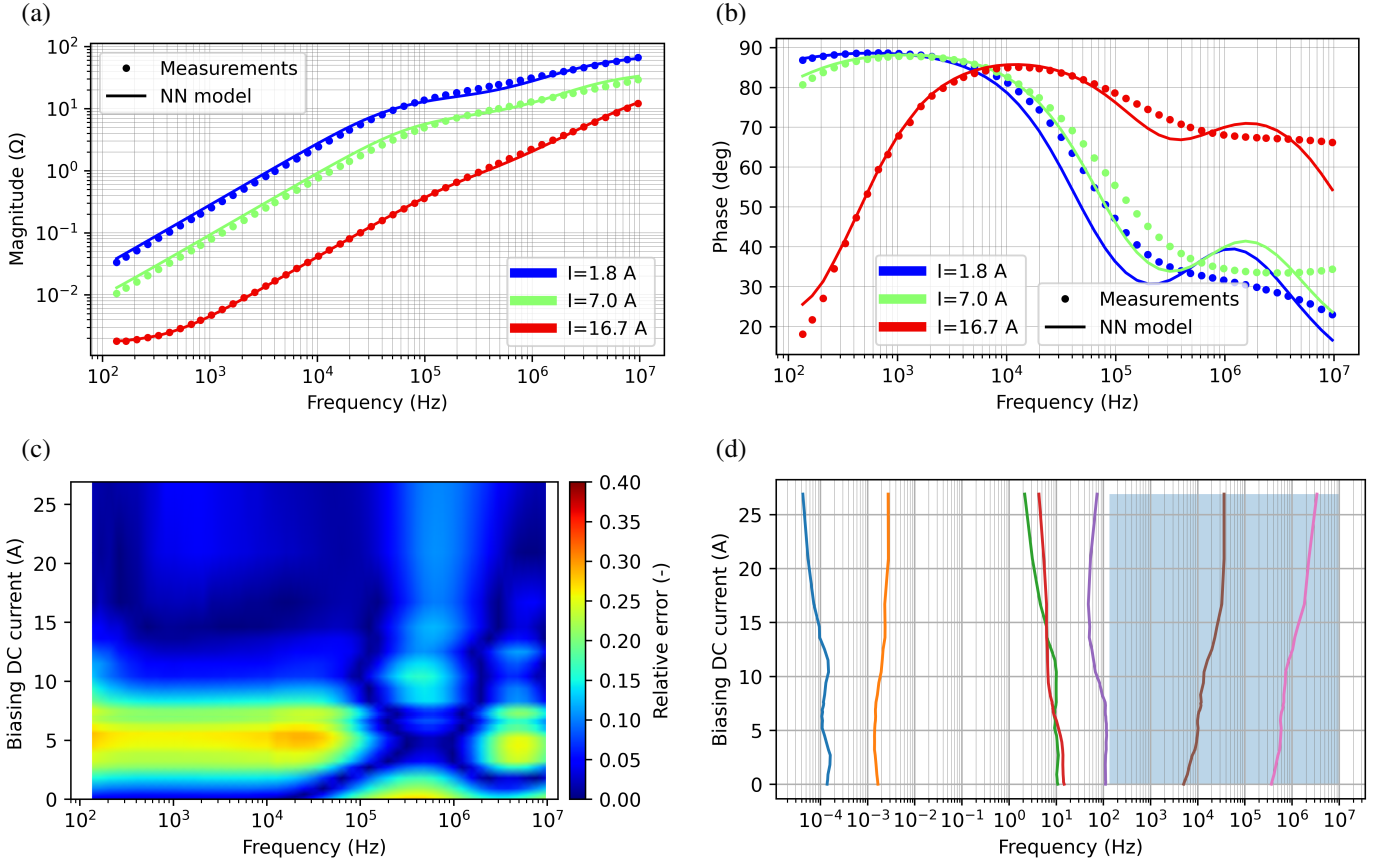
Figure 4. Results for testing the basic neural network model on the measured ring data: (a,b) examples of fitting the analytical model (solid curves), with parameters predicted by the NN model, versus measurements (dots) for the selected DC-bias currents $I_{\mathrm{DC}} = \{1.8, 7.0, 16.7\}$ A, (c) relative fitting error for the various DC-bias currents I_{DC} plotted versus frequency, and (d) characteristic frequencies $f_i = 2\pi R_i/L_i$ of the predicted NN model parameters (each line represent f_i for one LR pair in the LEEC series) versus DC-bias current I_{DC}. The blue color marks the frequency range where the measurements data are located.

In the same way, we define $\mathscr{L}_{\mathrm{cont}}(R)$. Moreover, we found that in our case it is beneficial to add yet another term to the loss function:

$$
l_{\mathrm{mono}}(L) = \begin{cases} 0 & \text{if } \frac{1}{N_{\mathrm{bias}}-1}\sum_{k=1}^{N_{\mathrm{bias}}-1}(L_{k+1}-L_k) < 0, \\ \frac{1}{N_{\mathrm{bias}}-1}\sum_{k=1}^{N_{\mathrm{bias}}-1}(L_{k+1}-L_k) & \text{otherwise,} \end{cases}
\tag{8}
$$

in order to maintain a monotonic decrease in the parameter values with the DC-bias current I_{DC}, which is justified by the ring material saturation. The total monotonicity loss $\mathscr{L}_{\mathrm{mono}}(L)$ is defined in the same way as in Eq. 7. All in all, the loss function for the Siamese NN model is as follows:

$$
\mathscr{L}_{\mathrm{Siam}} = \mathscr{L}_{\mathrm{dec}} + \alpha_1\mathscr{L}_{\mathrm{cont}}(L) + \alpha_2\mathscr{L}_{\mathrm{cont}}(R) + \\ + \alpha_3\mathscr{L}_{\mathrm{mono}}(L) + \alpha_4\mathscr{L}_{\mathrm{mono}}(R),
\tag{9}
$$

with the training hyperparameters α_i that control the impact of the loss components responsible for keeping the continuity of the parameters under I_{DC} changes.

If we look at the results for the Siamese NN model presented in Fig. 6(a,b), where we force the model to predict similar parameters for similar characteristics, we can observe that now characteristic frequencies $\omega_i = R_i/L_i$ change with the DC-bias current I_{DC} continuously and in some case monotonically

– see Fig. 6(b). Moreover, a surprising but also very desirable property is that forcing the continuity for the change of the parameters $\{L_i, R_i\}$ with I_{DC} at the same time has made fitting of the analytical model better than before – in Fig. 6(a) the relative fitting error typically does not exceed 15% with the average value over frequencies and currents equals 7.2%.

VI. MODIFIED SIAMESE MODEL

Fig. 6(b) presents yet another feature of the Siamese model (which was also partially present in the basic NN model) that it has a strong tendency to select features-parameters of the predicted analytical model (Eq. 1). It manifests in the fact that some groups of parameters, i.e. LR pairs in a series, are irrelevant – their characteristic frequencies, here ω_1, ω_2, ω_3, lie outside the ring working frequencies, which means that they can be safely skipped in the model. This motivates us to reduce the dimensionality of the model by assuming $M = 5$. Moreover, we observed that enforcing a more uniform positioning of the characteristic frequencies in the frequency domain over which the model operates can be beneficial in terms of further reducing the model size. Therefore, we added
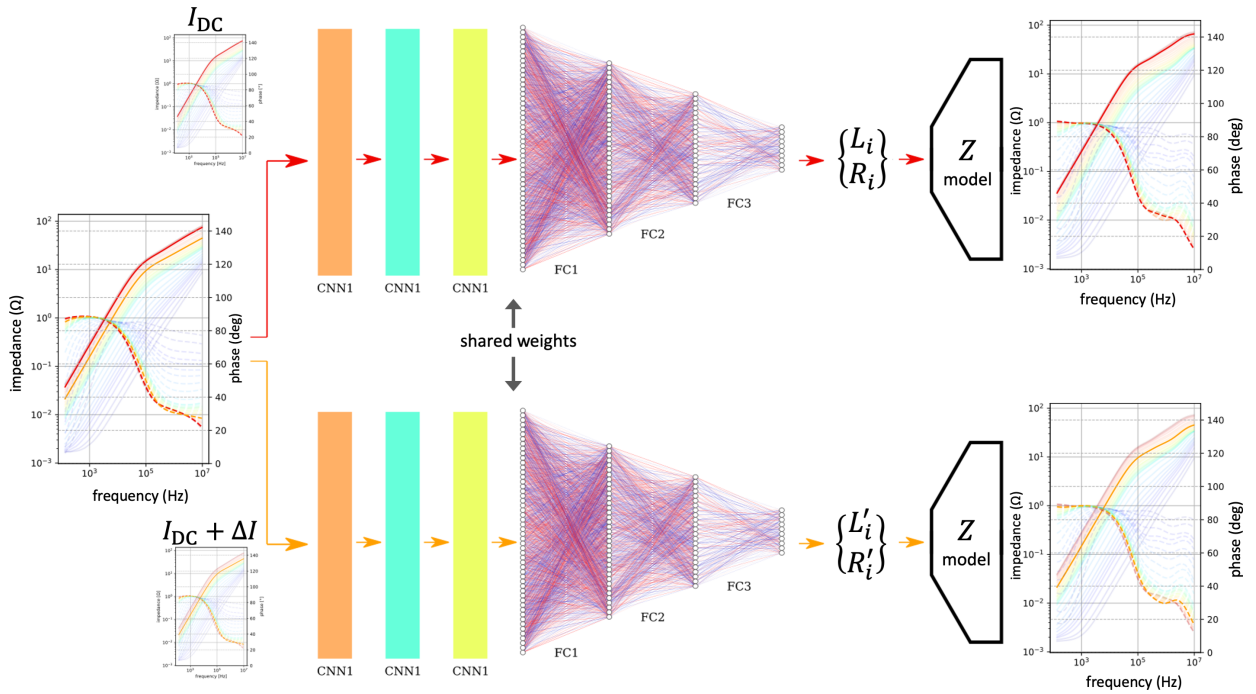
Figure 5. Siamese neural network model composed of two copies of the basic model that share the weights and process two neighbor characteristics (that differ in DC-bias current by ΔI) at the same time enabling to train the model to predict similar (or continuously changing) parameters for similar characteristics.

another component to the loss function of the form:

$$\mathscr{L}_\omega = \frac{1}{NM} \sum_{j=1}^{N} \sum_{i=1}^{M} \left(\frac{R_i}{L_i} - \Omega_i \right)^2, \tag{10}$$

with $N = N_{\text{train}} \times N_{\text{bias}}$, $M = 5$, and the constant frequency nodes centered at positions $\Omega_i = \{10^3, 10^{4.75}, 10^{6.5}, 10^{8.25}, 10^{10}\}$ Hz. The values of Ω_i have been chosen to evenly cover the frequency range of the measurement data. Moreover, including an extra point above this range further improved the fitting. Now the loss function for the modified Siamese NN model is the following:

$$\mathscr{L}_{\text{Siam}'} = \mathscr{L}_{\text{Siam}} + \alpha_5 \mathscr{L}_\omega. \tag{11}$$

The results in Fig. 6(c,d) clearly show that now the positioning of the characteristic frequencies is more homogeneous and all of ω_i occupies ring working frequency range. In addition, the relative fitting error obtained here is less than 10% in most cases and on average is the smallest among all three NN models, reaching 5.2%.

VII. RESULTS DISCUSSION AND COMPARISON WITH OTHER METHODS

In the first approach to the problem, we defined a network that predicts the parameters directly (containing the encoder part only) and compares them with the parameters used to generate a given characteristic. Such training was strictly supervised by using the encoder loss, defined as:

$$\mathscr{L}_{\text{enc}} = \frac{1}{2NM} \sum_{j=1}^{N} \sum_{y_i \in \{L_i, R_i\}}^{2M} \log^2 \left(\frac{y_{ij}}{\hat{y}_{ij}} \right). \tag{12}$$

The sum in Eq. 12 goes over the training samples and parameters: $\hat{y}_{ij}$ means the i-th parameter (L_i or R_i) predicted for j-th sample, Z_j, whereas y_{ij} represents ground truth parameters – we already know them from the generation process. However, this approach simply failed because it was not possible to effectively train a neural model that would generalize beyond a limited set of characteristics being some narrow part of the training data.

Much better results came from the unsupervised training where the AE model is simply given a variety of characteristics and has to attempt to reconstruct them. The model proposed in our scheme is able to generalize well, which is shown in Table I where the validation error reaches a few percent for the training set with a huge variety of characteristics. Consequently, the NN model is able to correctly fit parameters for previously unseen measured test data. At the same time, it is worth noting that the diversity of the training set (see Fig.2) fulfilled a kind of regularization, ensuring that the NN does not overfit any particular behavior.

In the proposed approach, the decoder is needed only to estimate the loss function ($\mathcal{L}_{\text{dec}}$), that is minimized during the training. The only requirement is that the decoder formula, which will pass the fitting error when the encoder is trained by backpropagation, should be differentiable. The approach where a large component of NN model (in our case, the decoder) is used only to estimate the error of the predictive part (encoder) in ML is not unusual. It is characteristic, for example, for generative adversarial networks (GANs) [49], where the discriminator is trained to estimate the error of the generator, or in neural style transfer models [50], where the pretrained NN determines so-called style or content losses.

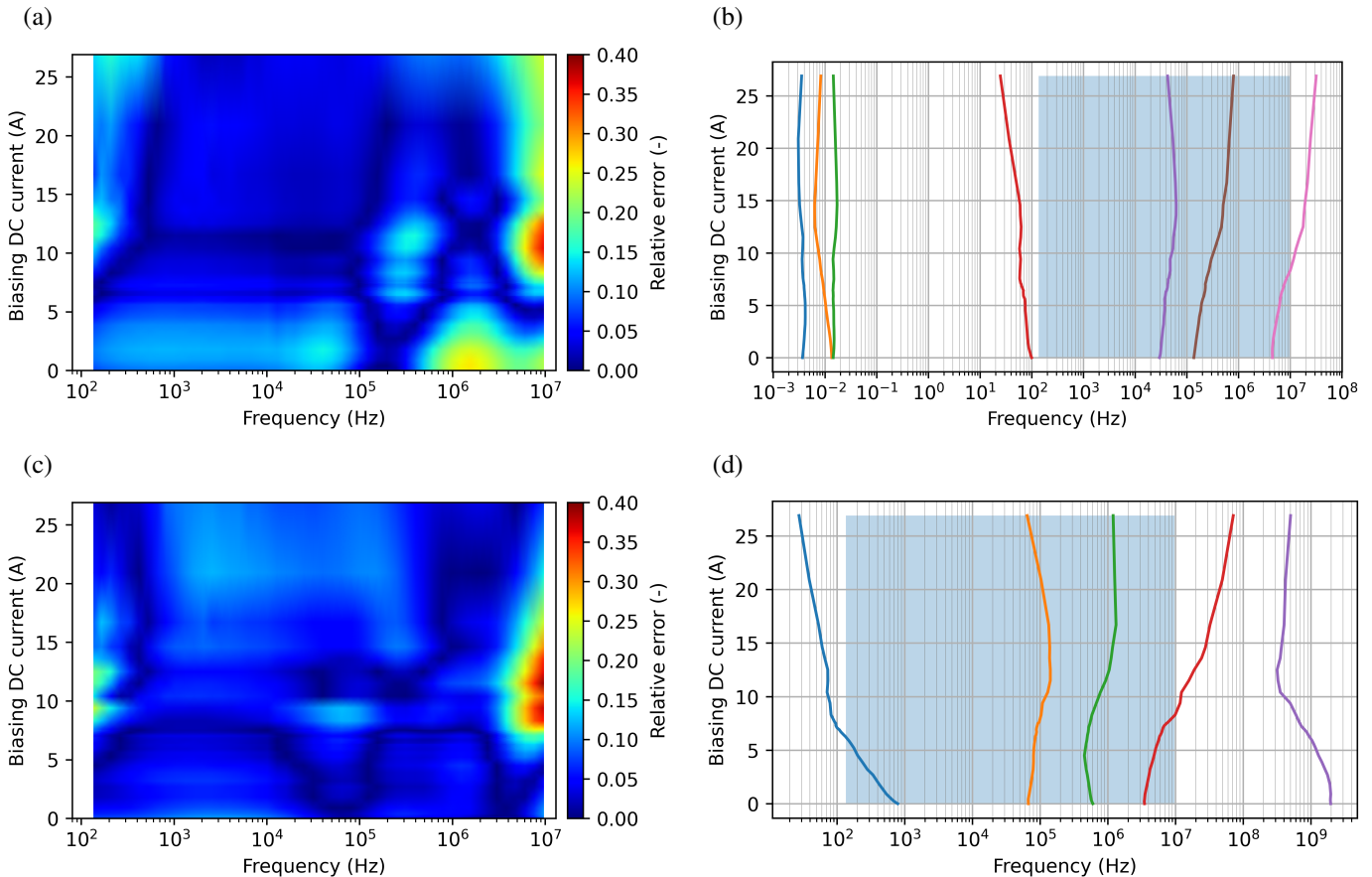In standard methods for finding equivalent circuit elements,

Figure 6. Same as in Fig. 4 but results for (a,b) Siamese neural network, and (c,d) modified Siamese model.

there is difficulty in capturing the correlation between individual $Z(f)$ characteristics for different DC-bias currents [41], and they are usually only able to match individual characteristics [51], [52]. This results in discontinuous parameters (here, the ladder elements L, R) as a function of the DC current. The Siamese network proposed here solves this problem.

Table I also shows the results following Refs. [42, 41], with fitting the lumped model to similar characteristics using other methods, including standard (nonlinear) least squares (NLS) and differential evolution (DE) [42], or modified NLS [41]. It is clear that NLS method does not handle the given problem, while DE gives comparable but worse results than Siamese networks. Modified NLS gives result comparable to Siamese NN, however, one should note that it uses constant $\omega_i = R_i/L_i$ [41], which is less general than NN, where L_i and R_i are searched independently. Moreover, modified NLS requires heavy grid search, which greatly increases the fitting time. On the other hand, the time required to train NN models is significantly shorter than the typical fitting process. Furthermore, once the NN is trained, making predictions for any (albeit similar to the training set) characteristics takes only a few seconds.

Among other approaches, vector fitting [52] which is a standard method for finding a transfer function parameters, not only has the problem of ensuring continuous parameters as a function of DC current but also cannot guarantee that

the identified transfer function is a positive real function [53]. Using neural networks allows for the imposition of arbitrary constraints and boundary conditions, including enforcing the physical behavior of the fitted parameters.

The proposed method exhibits a robust tendency to parameter selection. Although the initially suggested complexity of analytical model was $M = 7$, the NN network identified some components as having negligible contributions (as shown in Fig. 5(b)), effectively reducing the complexity $M \to 4$. Consequently, in the subsequent experiment (Fig. 5(d)), NN successfully predicted a simplified analytical model with $M = 5$ while maintaining a comparably low error.

One way to further improve the presented model is to eliminate the arbitrarily determined grid of Ω_i (Eq. 10) by adding a term in loss function that penalizes the characteristic frequencies of the individual LR ladder elements coming too close to each other. This will result in a more uniform distribution of these frequencies without imposing their positions from above.

VIII. METHODS

Data generation was performed with the usage of Pytorch library [54] on GPU (RTX A6000 48GB GDDR6), which significantly speedups the generation process from a few hours (when using CPU – 64 cores 128 threads Ryzen Threadripper 3990X) to a dozen seconds (GPU). NN model (encoder

and decoder part) was implemented using Pytorch with an automatic differentiation engine (autograd) used to track the gradient of the loss with respect to the model parameters during the backpropagation stage. The training of the single model on the GPU Nvidia RTX A6000 took about 1 hour.

Measurements were taken for NANOPERM nanocrystalline rings from Magnetec GmbH labeled M-616 [55], for different magnetization currents I_{DC} up to ring saturation using the Keysight Technologies Impedance Analyzer E4990A-030 to measure the impedance frequency characteristic and Keysight Technologies DC Power Supply N8731A to supply the DC-bias current I_{DC}.

IX. CONCLUSIONS

We proposed an unsupervised machine learning method, based on an autoencoder neural network, to estimate parameters of lumped element models for describing power magnetic material characteristics. The proposed method is general and in principle can be applied to any analytical model. The decoder introduces the analytical formula which parameters are predicted by the encoder. The decoder is needed to estimate the loss function, that is minimized during the training process.

In the proposed framework the materials are represented by multidimensional features, in our case as normalized impedance data in the form of real and imaginary parts. The way the neural network is trained is designed so that we can naturally enforce continuous (or any other physical) behavior on the entire family of measurement characteristics simultaneously. Once trained, the neural network can predict parameters for various characteristics in a few seconds, which is a different approach from the standard fitting procedure. In the paper, we discuss various architectures of neural networks to solve this problem, showing that the Siamese configuration of autoencoders gives the best results and makes it easier to impose key physical constraints when correlating a whole family of measured characteristics. We discuss the proposed method in comparison to previously studied methods.

During the training, the autoencoder is provided with a variety of synthetically generated characteristics and tasked with reconstructing them. Our method demonstrates strong generalization capabilities, as shown in Table I, where the validation error reaches a few percent for the training set with a huge variety of characteristics. Consequently, the neural network model is able to correctly fit parameters for previously unseen measured test data. Moreover, the method can accurately capture the correlation between individual characteristics for different DC-bias currents, resulting in continuous behavior of the parameters (i.e. the ladder elements, L and R) as a function of the DC current, which along with the good fitting accuracy demonstrates a key capability of the method.

X. ACKNOWLEDGMENTS

REFERENCES

[1] H. Li, D. Serrano, T. Guillod, S. Wang, E. Dogariu, A. Nadler, M. Luo, V. Bansal, N. K. Jha, Y. Chen *et al.*, "How magnet: Machine learning framework for modeling power magnetic material characteristics," *IEEE Transactions on Power Electronics*, 2023.

[2] D. Serrano, H. Li, S. Wang, T. Guillod, M. Luo, V. Bansal, N. K. Jha, Y. Chen, C. R. Sullivan, and M. Chen, "Why magnet: Quantifying the complexity of modeling power magnetic material characteristics," *IEEE Transactions on Power Electronics*, 2023.

[3] H. Li, D. Serrano, S. Wang, and M. Chen, "Magnet-ai: Neural network as datasheet for magnetics modeling and material recommendation," *IEEE Transactions on Power Electronics*, 2023.

[4] E. Polak, *Optimization: algorithms and consistent approximations*. Springer Science & Business Media, 2012, vol. 124.

[5] S. Kirkpatrick, C. D. Gelatt Jr, and M. P. Vecchi, "Optimization by simulated annealing," *science*, vol. 220, no. 4598, pp. 671–680, 1983.

[6] M. R. Bonyadi and Z. Michalewicz, "Particle swarm optimization for single objective continuous space problems: A review," *Evolutionary Computation*, vol. 25, no. 1, pp. 1–54, 2017.

[7] J. Kennedy and R. Eberhart, "Particle swarm optimization," in *Proceedings of ICNN'95 - International Conference on Neural Networks*, vol. 4, 1995, pp. 1942–1948 vol.4.

[8] K. Price, R. M. Storn, and J. A. Lampinen, *Differential evolution: a practical approach to global optimization*. Springer Science & Business Media, 2006.

[9] A. E. Eiben and J. E. Smith, *Introduction to evolutionary computing*. Springer-Verlag Berlin Heidelberg, 2015.

[10] J. Chai, H. Zeng, A. Li, and E. W. Ngai, "Deep learning in computer vision: A critical review of emerging techniques and application scenarios," *Machine Learning with Applications*, vol. 6, p. 100134, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S2666827021000670

[11] M. Alam, M. Samad, L. Vidyaratne, A. Glandon, and K. Iftekharuddin, "Survey on deep neural networks in speech and vision systems," *Neurocomputing*, vol. 417, pp. 302–321, 2020. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231220311619

[12] B. Selvaratnam and R. T. Koodali, "Machine learning in experimental materials chemistry," *Catalysis Today*, vol. 371, pp. 77–84, 2021.

[13] P. M. Voyles, "Informatics and data science in materials microscopy," *Current Opinion in Solid State and Materials Science*, vol. 21, no. 3, pp. 141–158, 2017.

[14] K. T. Schütt, M. Gastegger, A. Tkatchenko, K.-R. Müller, and R. J. Maurer, "Unifying machine learning and quantum chemistry with a deep neural network for molecular wavefunctions," *Nature communications*, vol. 10, no. 1, p. 5024, 2019.

[15] C. Chen, W. Ye, Y. Zuo, C. Zheng, and S. P. Ong, "Graph networks as a universal machine learning framework for molecules and crystals," *Chemistry of Materials*, vol. 31, no. 9, pp. 3564–3572, 2019.

[16] W. Li, P. Chen, B. Xiong, G. Liu, S. Dou, Y. Zhan, Z. Zhu, T. Chu, Y. Li, and W. Ma, "Deep learning modeling strategy for material science: from natural materials to metamaterials," *Journal of Physics: Materials*, vol. 5, no. 1, p. 014003, mar 2022. [Online]. Available: https://dx.doi.org/10.1088/2515-7639/ac5914

[17] A. S. Fuhr and B. G. Sumpter, "Deep generative models for materials discovery and machine learning-accelerated innovation," *Frontiers in Materials*, vol. 9, 2022. [Online]. Available: https://www.frontiersin.org/articles/10.3389/fmats.2022.865270

[18] K. Choudhary, B. DeCost, C. Chen, A. Jain, F. Tavazza, R. Cohn, C. W. Park, A. Choudhary, A. Agrawal, S. J. L. Billinge, E. Holm, S. P. Ong, and C. Wolverton, "Recent advances and applications of deep learning methods in materials science," *npj Computational Materials*, vol. 8, no. 1, p. 59, 2022. [Online]. Available: https://doi.org/10.1038/s41524-022-00734-6

[19] C. Bonatti and D. Mohr, "One for all: Universal material model based on minimal state-space neural networks," *Science Advances*, vol. 7, no. 26, p. eabf3658, 2021. [Online]. Available: https://www.science.org/doi/abs/10.1126/sciadv.abf3658

[20] J. Damewood, J. Karaguesian, J. R. Lunger, A. R. Tan, M. Xie, J. Peng, and R. Gómez-Bombarelli, "Representations of materials for machine learning," *Annual Review of Materials Research*, vol. 53, 2023.

[21] H. Saliah, D. Lowther, and B. Forghani, "A neural network model of magnetic hysteresis for computational magnetics," *IEEE Transactions on Magnetics*, vol. 33, no. 5, pp. 4146–4148, 1997.

[22] C. Serpico and C. Visone, "Magnetic hysteresis modeling via feed-forward neural networks," *IEEE Transactions on Magnetics*, vol. 34, no. 3, pp. 623–628, 1998.

[23] H. Saliah, D. Lowther, and B. Forghani, "Modeling magnetic materials using artificial neural networks," *IEEE Transactions on Magnetics*, vol. 34, no. 5, pp. 3056–3059, 1998.

[24] F. Sixdenier, R. Scorretti, R. Marion, and L. Morel, "Quasistatic hysteresis modeling with feed-forward neural networks: Influence of the last but one extreme values," *Journal of Magnetism and Magnetic Materials*, vol. 320, no. 20, pp. e992–e996, 2008, proceedings of the 18th International Symposium on Soft Magnetic Materials. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S030488530800543X

[25] Z. Zhao, F. Liu, S. L. Ho, W. N. Fu, and W. Yan, "Modeling magnetic hysteresis under dc-biased magnetization using the neural network," *IEEE Transactions on Magnetics*, vol. 45, no. 10, pp. 3958–3961, 2009.

[26] F. Riganti Fulginei and A. Salvini, "Neural network approach for modelling hysteretic magnetic materials under distorted excitations," *IEEE Transactions on Magnetics*, vol. 48, no. 2, pp. 307–310, 2012.

[27] S. Quondam Antonio, F. Riganti Fulginei, A. Laudani, A. Faba, and E. Cardelli, "An effective neural network approach to reproduce magnetic hysteresis in electrical steel under arbitrary excitation waveforms," *Journal of Magnetism and Magnetic Materials*, vol. 528, p. 167735, 2021. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0304885321000111

[28] C. Grech, M. Buzio, M. Pentella, and N. Sammut, "Dynamic ferromagnetic hysteresis modelling using a preisach-recurrent neural network model," *Materials*, vol. 13, no. 11, 2020. [Online]. Available: https://www.mdpi.com/1996-1944/13/11/2561

[29] H. Li, D. Serrano, S. Wang, and M. Chen, "Magnet-ai: Neural network as datasheet for magnetics modeling and material recommendation," *IEEE Transactions on Power Electronics*, vol. 38, no. 12, pp. 15 854–15 869, 2023.

[30] H. Li, D. Serrano, T. Guillod, S. Wang, E. Dogariu, A. Nadler, M. Luo, V. Bansal, N. K. Jha, Y. Chen, C. R. Sullivan, and M. Chen, "How magnet: Machine learning framework for modeling power magnetic material characteristics," *IEEE Transactions on Power Electronics*, vol. 38, no. 12, pp. 15 829–15 853, 2023.

[31] M. Ferch, "Application overview of nanocrystalline inductive components in today's power electronic systems," in *Proc. Soft Magn. Mater. Conf*, no. A1-01, 2013.

[32] U. Riechert, M. Bösch, M. Szewczyk, W. Piasecki, J. Smajic, A. Shoory, S. Burow, and S. Tenbohlen, "Mitigation of very fast transient overvoltages in gas insulated uhv substations," *Proc. CIGRE*, 2012.

[33] D. C. Jiles and D. L. Atherton, "Theory of ferromagnetic hysteresis (invited)," *Journal of Applied Physics*, vol. 55, no. 6, pp. 2115–2120, 03 1984. [Online]. Available: https://doi.org/10.1063/1.333582

[34] F. Liorzou, B. Phelps, and D. Atherton, "Macroscopic models of magnetization," *IEEE Transactions on Magnetics*, vol. 36, no. 2, pp. 418–428, 2000.

[35] L. Perkkiö, B. Upadhaya, A. Hannukainen, and P. Rasilo, "Stable adaptive method to solve fem coupled with jiles–atherton hysteresis model," *IEEE Transactions on Magnetics*, vol. 54, no. 2, pp. 1–8, 2018.

[36] K. Hoffmann, J. P. Assumpção Bastos, J. V. Leite, N. Sadowski, and F. Barbosa, "An accurate vector jiles-atherton model for improving the fem convergence," in *2016 IEEE Conference on Electromagnetic Field Computation (CEFC)*, 2016, pp. 1–1.

[37] M. Chen, M. Araghchini, K. K. Afridi, J. H. Lang, C. R. Sullivan, and D. J. Perreault, "A systematic approach to modeling impedances and current distribution in planar magnetics," *IEEE Transactions on Power Electronics*, vol. 31, no. 1, pp. 560–580, 2016.

[38] H. Dommel and W. Meyer, "Computation of electromagnetic transients," *Proceedings of the IEEE*, vol. 62, no. 7, pp. 983–993, 1974.

[39] H. W. Dommel and W. S. Meyer, "Computation of electromagnetic transients," *Proceedings of the IEEE*, vol. 62, no. 7, pp. 983–993, 1974.

[40] M. Szewczyk, K. Kutorasiński, J. Pawłowski, W. Piasecki, and M. Florkowski, "Advanced modeling of magnetic cores for damping of high-frequency power system transients," *IEEE Transactions on Power Delivery*, vol. 31, no. 5, pp. 2431–2439, 2016.

[41] K. Kutorasiński, J. Pawłowski, P. Leszczyński, and M. Szewczyk, "Nonlinear modeling of magnetic materials for circuit simulations," *Scientific Reports*, vol. 13, no. 1, p. 17178, 2023.

[42] J. Pawłowski, K. Kutorasiński, and M. Szewczyk, "Multifrequency nonlinear model of magnetic material with artificial intelligence optimization," *Scientific reports*, vol. 12, no. 1, pp. 19 784–19 784, 2022.

[43] K. Kutorasiński, M. Szewczyk, M. Molas, and J. Pawłowski, "Measuring impedance frequency characteristics of magnetic rings with dc-bias current," *ISA Transactions*, 2023. [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0019057823003968

[44] S. I. Nikolenko, *Synthetic data for deep learning (Springer Optimization and Its Applications)*. Springer, 2021, vol. 174.

[45] M. D. Zeiler and R. Fergus, "Visualizing and understanding convolutional networks," in *European conference on computer vision*. Springer, 2014, pp. 818–833.

[46] F. Yu and V. Koltun, "Multi-scale context aggregation by dilated convolutions," in *International Conference on Learning Representations (ICLR)*, May 2016.

[47] A. Vaswani, N. Shazeer, N. Parmar, J. Uszkoreit, L. Jones, A. N. Gomez, L. u. Kaiser, and I. Polosukhin, "Attention is all you need," in *Advances in Neural Information Processing Systems*, I. Guyon, U. V. Luxburg, S. Bengio, H. Wallach, R. Fergus, S. Vishwanathan, and R. Garnett, Eds., vol. 30. Curran Associates, Inc., 2017. [Online]. Available: https://proceedings.neurips.cc/paper_files/paper/2017/file/3f5ee243547dee91fbd053c1c4a845aa-Paper.pdf

[48] F. Schroff, D. Kalenichenko, and J. Philbin, "Facenet: A unified embedding for face recognition and clustering," *2015 IEEE Conference on Computer Vision and Pattern Recognition (CVPR)*, Jun 2015. [Online]. Available: http://dx.doi.org/10.1109/CVPR.2015.7298682

[49] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial networks," *Commun. ACM*, vol. 63, no. 11, p. 139–144, oct 2020. [Online]. Available: https://doi.org/10.1145/3422622

[50] J. Johnson, A. Alahi, and L. Fei-Fei, "Perceptual losses for real-time style transfer and super-resolution," in *Computer Vision – ECCV 2016*, B. Leibe, J. Matas, N. Sebe, and M. Welling, Eds. Cham: Springer International Publishing, 2016, pp. 694–711.

[51] M. Szewczyk, J. Pawlowski, K. Kutorasinski, S. Burow, S. Tenbohlen, and W. Piasecki, "Identification of rational function in s-domain describing a magnetic material frequency characteristics," in *9th IET International Conference on Computation in Electromagnetics (CEM 2014)*. IET, 2014, pp. 1–2.

[52] B. Gustavsen and A. Semlyen, "Rational approximation of frequency domain responses by vector fitting," *IEEE Transactions on power delivery*, vol. 14, no. 3, pp. 1052–1061, 1999.

[53] L. De Tommasi, D. Deschrijver, and T. Dhaene, "Single-input-single-output passive macromodeling via positive fractions vector fitting," in *2008 12th IEEE Workshop on Signal Propagation on Interconnects*. IEEE, 2008, pp. 1–2.

[54] A. Paszke, S. Gross, F. Massa, A. Lerer, J. Bradbury, G. Chanan, T. Killeen, Z. Lin, N. Gimelshein, L. Antiga *et al.*, "Pytorch: An imperative style, high-performance deep learning library," *Advances in neural information processing systems*, vol. 32, 2019.

[55] "Product specification for inductive components: M-616 core datasheet," *MAGNETEC GmbH*, Jan, 2020.