mathematics

MDPI

Article

Neural Network Architectures and Magnetic Hysteresis: Overview and Comparisons

Silvia Licciardi [1,*,†], Guido Ala [1,†], Elisa Francomano [1,†], Fabio Viola [1,†], Michele Lo Giudice [2,†], Alessandro Salvini [2,†], Fausto Sargeni [3,†], Vittorio Bertolini [4,†], Andrea Di Schino [4,†], Antonio Faba [4,†]

[1] Department of Electrical Engineering, University of Palermo, Viale delle Scienze, 90128 Palermo, Italy; guido.ala@unipa.it (G.A.); elisa.francomano@unipa.it (E.F.); fabio.viola@unipa.it (F.V.)
[2] Department of Civil, Computer Science and Aeronautical Technologies Engineering, University of Rome Tre, Via Vito Volterra 62, 00146 Rome, Italy; michele.logiudice@uniroma3.it (M.L.G.); alessandro.salvini@uniroma3.it (A.S.)
[3] Department of Electronic Engineering, University of Rome Tor Vergata, Via del Politecnico 1, 00133 Rome, Italy; fausto.sargeni@uniroma2.it
[4] Department of Engineering, University of Perugia, Via G. Duranti 93, 06123 Perugia, Italy; vittorio.bertolini@dottorandi.unipg.it (V.B.); andrea.dischino@unipg.it (A.D.S.); antonio.faba@unipg.it (A.F.)
* Correspondence: silvia.licciardi@unipa.it or silviakant@gmail.com
† The authors contributed to this work according to what declared in the Author Contribution Section.

Abstract: The development of innovative materials, based on the modern technologies and processes, is the key factor to improve the energetic sustainability and reduce the environmental impact of electrical equipment. In particular, the modeling of magnetic hysteresis is crucial for the design and construction of electrical and electronic devices. In recent years, additive manufacturing techniques are playing a decisive role in the project and production of magnetic elements and circuits for applications in various engineering fields. To this aim, the use of the deep learning paradigm, integrated with the most common models of the magnetic hysteresis process, has become increasingly present in recent years. The intent of this paper is to provide the features of a wide range of deep learning tools to be applied to magnetic hysteresis context and beyond. The possibilities of building neural networks in hybrid form are innumerable, so it is not plausible to illustrate them in a single paper, but in the present context, several neural networks used in the scientific literature, integrated with various hysteretic mathematical models, including the well-known Preisach model, are compared. It is shown that this hybrid approach not only improves the modeling of hysteresis by significantly reducing computational time and efforts, but also offers new perspectives for the analysis and prediction of the behavior of magnetic materials, with significant implications for the production of advanced devices.

Keywords: deep learning; LSTM architectures; hybrid neural networks architectures; magnetic hysteresis; Preisach model; numerical methods; global optimization; gradient methods

MSC: 68T07; 68T99; 78A25; 78A99; 65K05; 90C52

Citation: Licciardi, S.; Ala, G.; Francomano, E.; Viola, F.; Lo Giudice, M.; Salvini, A.; Sargeni, F.; Bertolini, V.; Di Schino, A.; Faba, A. Neural Network Architectures and Magnetic Hysteresis: Overview and Comparisons. *Mathematics* **2024**, *12*, 3363. https://doi.org/10.3390/math12213363

Academic Editor: Ke-Lin Du

Received: 26 September 2024
Revised: 22 October 2024
Accepted: 24 October 2024
Published: 26 October 2024

1. Introduction

Magnetic hysteresis is a fundamental phenomenon in electromagnetism describing the non-linear behavior of ferromagnetic materials [1–3]. There are several mathematical models to represent the trend. The Jiles–Atherton model [4–8], for example, describes the phenomenon using a physical approach based on energy considerations and magnetization theory. It is particularly known for its ability to model real magnetic materials with good precision, when there is no significant complexity in the hysteresis processes. The Stoner–Wohlfarth model [9,10] describes hysteretic behavior in ferromagnetic nanoparticles and granules. It assumes that magnetization rotates consistently within a particle, making it

particularly useful for small systems. The Prandtl–Ishlinskii model [11,12] is also used for other types of hysteresis besides magnetic ones. It is known for its simplicity and ability to model rate-independent hysteresis. Many other models can be mentioned, such as those of Krasonsel'skii–Pokrovskii, Maxwell, Bouc–Wen, Dahl and so on [13], but of all, certainly, the most used is the Preisach model (PM) [10,14,15] which represents one of the most effective methods of describing hysteresis, using an overlap of elementary operators to capture the complexity of such phenomena. Since its initial formulation, various models have been developed to include other particular characteristics of the process under analysis. We mention only some of the most relevant: (1) Non-linear Generalized PM (GNP) [10,13,16–18] allows hysteresis to be described more flexibly, being more accurate even when considering magnetic materials whose non-linear properties do not satisfy the classic PM. (2) Time-dependent PM [10,16,19], by introducing the temporal dependence, allows to describe phenomena of dynamic hysteresis, where the system response depends on the rate of change in the applied field. (3) Stochastic PM [10,20,21] includes random elements to represent uncertainty and variability in materials. This extension is useful for modeling systems in which magnetic behavior presents intrinsic noise or variability due to factors such as material imperfections, thermal fluctuations or other forms of stochastic disturbances. (4) For PM with limited memory [10,21,22], only part of the past history affects the current state of the system. This approach is particularly relevant for systems with short-term memory. (5) Inverse PM [16,20] is used to identify material characteristics from system response measurements. This approach is useful for the characterization and experimental analysis of materials. (6) Vectorial PM [23,24] extends the original model to include vector dependence, allowing to describe three-dimensional systems and their response to magnetic fields applied in different directions. (7) PM with Interaction between Hysterons [25–28] is an extension that introduces the interaction between hysterons, making the model more realistic for materials in which the elementary units of hysteresis are not independent but affect each other's response.

1.1. Preisach Model

Among the mentioned hysteretic models, therefore, the Preisach model deserves particular attention. First, its conceptually simple but highly flexible nature makes it a powerful tool for describing hysteresis in a wide range of materials and operating conditions. It is based on the idea that hysteresis can be represented as the sum of elementary units called hysterones, each of which responds to the magnetic field with a non-linear behavior. This modular approach allows complex phenomena, such as material history dependence and cyclic behavior, to be modeled with precision. Another aspect that makes it particularly relevant is its ability to adapt to magnetic systems with different physical properties, without requiring detailed knowledge of the microscopic processes involved, which makes it ideal for practical applications, where materials can exhibit highly non-linear and variable behavior, often difficult to capture with simpler physical models. Finally, the PM is distinguished by its computational flexibility. Due to its scalable nature, it can be implemented in numerical simulations and control algorithms for complex magnetic systems, ensuring a good compromise between accuracy and speed of calculation. Therefore, among the various models of hysteresis that will be discussed in the present context, the Preisach model represents a milestone in modeling this phenomenon and will continue to play a central role both in theoretical research and engineering applications.

A hysteresis function represents the input/output connections with multi-branch non-linearities, where the turning points of the hysteresis branches are affected by the past minimum/maximum input values [1]. Such a definition outlines a key property of the hysteresis phenomenon, namely, a sequential behavior in which the output must be determined depending on both the input and internal states. PM can be used as a mathematical tool to describe this process [15] and can be built as follows.

Definition 1 (Preisach Model). *Let input $H(t)$, output $M(H)$ be uniquely determined:*

$$M(H) = \int\int \rho(H_{min}, H_{max})\hat{G}_{H_{min},H_{max}}(H(t))dH_{min}dH_{max}$$
$$\hat{G} = \begin{cases} +1 & H \geq H_{max} \\ -1 & H \leq H_{min} \\ H_{prev} & H_{min} < H < H_{max} \end{cases} \quad (1)$$

where H_{min} and H_{max} are the switching values up and down input, $\hat{G}_{H_{min},H_{max}}$ is the finite set of binary hysteresis operators useful as local memory, and $\rho(H_{min}, H_{max})$ are the hysterons distribution (model coefficients by experimental data).

1.2. Artificial Neural Network Architectures

As in many other fields of pure and applied sciences, the use of deep learning (DL) neural networks (NNs) is beneficial for the phenomenon of magnetic hysteresis as well. In recent years, applications of artificial neural networks (ANNs) to hysteresis models (basic or generalized) have shown considerable potential in improving the accuracy and efficiency of modeling [6,23,25–42]. Neural architectures can obviously be of various types, depending on the specifics of the problem to be treated and the objectives that are proposed, but the common result is always to have a substantial gain in terms of computational costs or memory allocation. Sometimes, this can be at the cost of better results in terms of accuracy or error compared to traditional methods [25]. To cope with this, we also resort to hybrid forms of ANN or combined forms of ANN and traditional models of hysteresis, as in the examples we will see below. In any case, assessments must of course be made on a case-by-case basis.

1.3. Methodology

In the following sections, we will show just a few of the several possible ANN, based on the results obtained in the literature and comparing the advantages or limitations of the various approaches. In particular, in Sections 2 and 3, we examine the use of feedforward neural networks (FNNs) and recurrent neural networks (RNNs) [42] in the context of modeling magnetic hysteresis by using the Preisach model. These peculiar neural networks deserve special attention because they are the basis of a multitude of hybrid forms. They are widely used in the literature for the characteristics which we will discuss in the following. In Section 4, we will provide a list of additional architectures used, summarizing the salient points. We will analyze the effectiveness of each type of network in predicting hysteretic behavior, comparing their performance in terms of accuracy, convergence rate, and generalization ability. The results show that the integration of neural networks with the Preisach model not only improves the representation of hysteresis but also offers new perspectives for the analysis and prediction of the behavior of magnetic materials. The approach to this research has required the study of a wide range of articles concerning neural networks both from a purely theoretical point of view and applied to the context of interest. The paper that has been carried out from this aims to provide flexibility in the choice of neural tools to be able to adapt them to the specific experiments while being aware of the costs and benefits of each technique. Our research highlights the importance of choosing the appropriate neural architecture for specific hysteresis applications, thus providing guidelines for future studies and practical applications.

2. Feedforward Neural Network

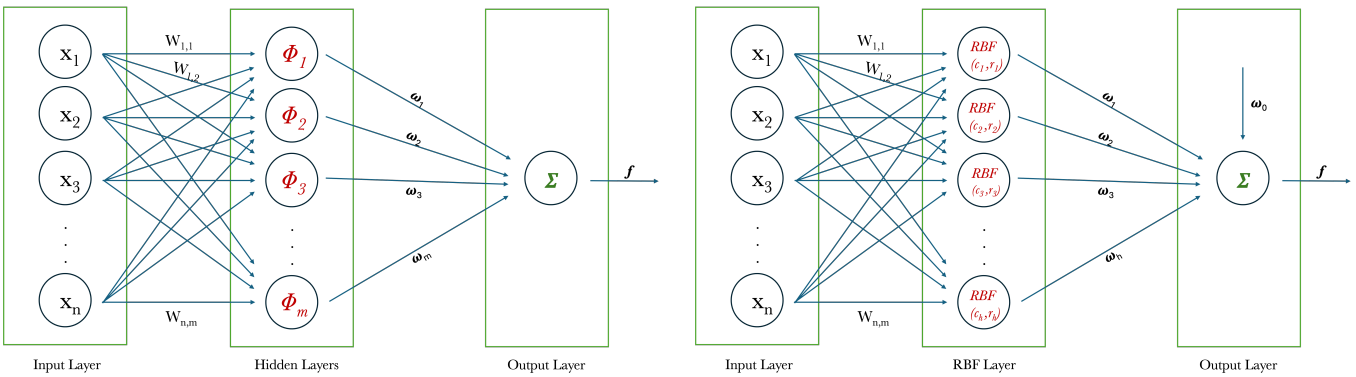
The first article that we will take into consideration treats the usage of ANN to implement the Preisach model for the modeling of magnetic cores [30]. Traditionally, this process requires complex and computationally expensive mathematical models. The authors propose instead to use the ANN to create a more flexible and adaptable model that is useful to simulate complex multi-variable and time-dependent processes. Like other neural networks, they can be used for specific cases as well as for general descriptions,

being able to predict even random behaviors of the system as well as to respond to inputs not belonging to the initial datasets. In FNN, among the most common NNs, information moves in one only direction, from the input to the output phase, without cycles or loops. The goal of the authors is that the network learns the complex relationship between the applied magnetic field H and the consequent magnetic induction B in the material. Once trained, neural networks are able to predict the behavior of the magnetic hysteresis cycle for different operating conditions without the need for detailed mathematical models. This approach offers a more efficient and flexible alternative to traditional methods.

Among the FNNs, in particular, the authors in [30] use the Radial Basis Function NN (RBF-NN), employed for classification, approximation problems and strict interpolation in multi-dimensional space [43–46]. They are preferred over multi-layer perceptron (MLP) because they are faster in the learning phase. This is due to the fact that this type of architecture does not require multi-layers but only one layer that includes the RBF, each of which depends on the respective centers c and amplitude r. The number of centers (and therefore of the respective connected neurons) can be high and the choice random, and this would imply high computational complexity and cause numerical ill-conditioning. To avoid this, the type of RBF and the type of centers should be carefully chosen. They are usually chosen from the points of the dataset (x_i), but this does not guarantee a complete mapping of the sample; instead, the approach used in [46], based on the Orthogonal Least Square (OLS) method, ensures better performance and the ability of the network to achieve the results of an MLP despite its two layers. The technique is based on the fact that RBFs are seen as special cases of a regression model

$$l(t) = \sum_{k=1}^{m} p_i(t)\delta_i + \varepsilon(t) \tag{2}$$

where $l(t)$ is the desired output, δ_i are the parameters, $p_i(t)$ is the so-called regressors (fixed functions of the input $x(t)$), and $\varepsilon(t)$ is the error assumed to be uncorrelated with $p_i(t)$. Through this formulation, in our context, a fixed center c_i with a non-linear activation function ϕ_i corresponds to a regressor $p_i(t)$ and the problem of how to select an appropriate set of RBF centers corresponds to the selection of considerable regressors from a given candidate set (OLS method). The iteration ends at step $M_s : \left(1 - \sum_{k=1}^{M_s} \varepsilon_k\right) < \rho$, with $\rho \in (0,1)$ as a chosen tolerance. In Figure 1a, a comparison is made between a classical MLP scheme and an RBF scheme.



(**a**) MLP scheme. (**b**) RBF scheme.

Figure 1. Comparison through (**a**) MPL and (**b**) RBF layers schemes.

Despite having a similar structure composed of input, hidden, and output layers, the activation functions are the radial basis, and therefore typically Gaussian, spline, multi-quadratic, or multi-quadratic inverse, instead of typically sigmoid. The included two parameters, c and r, associated with RBF neurons, are similar to hidden layer weights and biases in MLP. Finally, the output returns a weighted sum of its inputs using the appropriate

weights ω. The network training consists largely of what has been illustrated so far, namely, finding an adequate number of these parameters and associated neurons and calculating their weights through various algorithms. The OLS method to minimize the errors between the desired output and that predicted by the neural network is used by the authors in [30]. The expression for the processing of the input through function of activation and the restitution of the relative outputs has the standard form for the neural nets

$$f(x_1, \ldots, x_n) = \sum_{k=1}^{m} w_k \phi(|| \, x_k - c_k \, ||), \qquad \phi(x - c) = \exp\left(\frac{(x-c)^2}{r^2}\right). \tag{3}$$

where ϕ is the activation function, in this case, of the Gaussian type, and w_k is the weights. The correspondence with Equation (2) is evident. At this point, reconstruction of the Preisach model is obtained by using normalized hysteresis cycles data for the square Permalloy 80 at various frequencies.

The NN efficiency criteria are usually based on the number of epochs, training time, network size, dataset, loss function, and accuracy. For hysteresis modeling, the major criterion of interest is the *accuracy*. In [30], comparisons are made between experimental data and network output. To measure the error index, the Mean Square Error (MSE) of the normalized hysteresis modeling from the standard deviation of the experimental data (also called non-dimensional index error (NDIE)) is used. Error indices vary from 4.09% to 5.56% with an average NDIE of 4.63% (a comparison between the results obtained through the experimental data and the RBF outputs is shown in [30], Figure 3).

3. Hybrid Architectures—FNN and RNN

Also in [25,26,28], the authors use FNNs and PM to simulate hysteresis cycles in different materials, as well as hybrid models that will be discussed below (in such hybrid models, FNNs are used to calculate the memory-free relationship between input and output, while to take into account the memory effect, which is typical of hysteretic behavior, a hysteron-based model is adopted. This is to deal with the problem of the formulation of adequate dependence on the memory of the output model depending on the hysteretic behavior of magnetic materials. Instead of hybrid techniques, also approaches that are full network-based can be used, such as recurrent neural network architectures (RNNs), having an intrinsically recursive memory, as we will see in Section 4.1.). In particular, in [25], soft ferromagnetic materials (commercial iron-silicon NGO), subjected to sinusoidal and non-sinusoidal magnetic induction waveforms, are considered. The latter can lead to strongly distorted hysteresis cycles and, therefore, the design of the relative devices can be difficult. The proper selection of materials to be used for specific applications can be finalized using proper tools. Typically, designers rely on Finite Element Method (FEM) approaches, which are able to provide reliable predictions of the material performance when they work under given working conditions. In this analysis, the authors use PM to generate a larger dataset consisting of a family of first-order inversion curves (FORCs), suitable for NN training. The hysteresis model thus generated has the ability to also detect sub-loops in the cycles. The comparison between PM and NN is extended also to other measurements; in particular, several hysteresis cycles, taken for different kinds of excitation, are considered. Also in these cases, NN performs surprisingly well from the computational efforts and memory request sides. It should also be noted that the proposed method allows the reversal of the problem ($B \leftrightarrow H$) and therefore the comparison with the FEM (open problem). In the article, a first overview of the hysteresis models usually used is also proposed. In addition to the PM then used for comparisons, the play model, stop model, and those derived from Stoner and Wohlfarth are also considered, all very accurate but computationally expensive models. Conversely, models such as the Jiles–Atherton model are inexpensive but not as accurate. Unlike what is shown in [30], the optimization model for the output layer of the FNN used in [28] is the Levenberg–Marquardt algorithm [47]. The FNNs thus obtained are usable for sinusoidal magnetization processes as well as other kinds of waveforms, with the limitation related to the impossibility about the reproduction of sub-loops, for which a

technique called "transplantation" [48] is considered. As explained in [49], this algorithm is able to close a sub-loop operating a transplantation of the points in one branch of the sub-loop. An example is provided in Figure 2, where the blue branch is obtained from the corresponding points in the green branch.

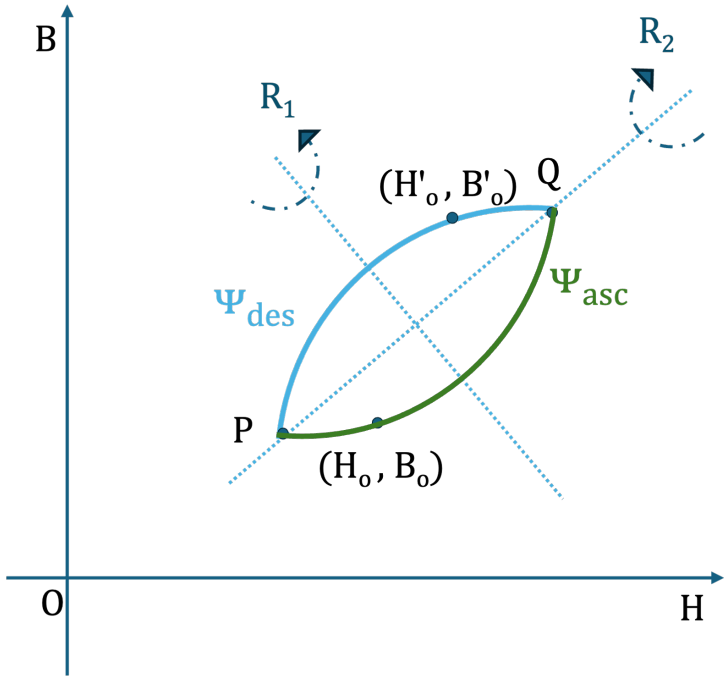


Figure 2. Transplantation technique. Blue branch is obtained from the corresponding points in the green branch.

To conclude, approaches based on NN are able to recognize and associate the proper hysteresis cycle to the specific analyzed sample.

Considering, therefore, the PM, there is the possibility to derive an analytical expression for the Preisach distribution by solving the Everett integral [10,50] numerically through various numerical methods ([51,52]), or by approximating with suitable probability density functions. To reduce the number of parameters required by the model, in [25] a formulation related to the Lorentzian probability density function approximation is proposed (in Figure 3, Gauss and Lorentz distributions are compared; the latter is chosen because of its slower diffusion). Recall that the Everett integral is known in the following form $M(t) = \iint_T P(U,V)dUdV - \iint_{-T} P(U,V)dUdV$, where $M(t)$ is the sum of the hysterons magnetization, and T and $-T$ are the positive and negative domains of the Preisach triangle, where hysterons contribute positively or negatively to the overall time-dependent magnetization [53].

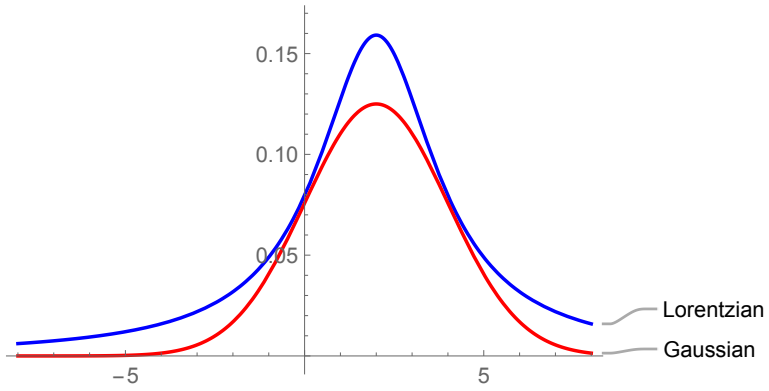


Figure 3. Comparison between Lorentzian and Gaussian distributions.

The couple H_i, u (interaction field and coercive field, respectively) identifies the hysterons which must be distributed in relation to both. The distribution function $P(H_i, u)$ can be written by using the principle of variables separation, thus obtaining

$$P(H_i, u) = \frac{\sigma_H \sigma_u}{\pi^2 (H_1^2 + \sigma_H^2)[(u - u_0)^2 + \sigma_u^2]} \tag{4}$$

with σ_H, σ_u parameters of the control of emissions H_i and u, u_0 being the most likely coercive field of hysterons (related to the material coercive field). An optimized array, structured according to specific rules, drives the disposition of hysterons on the magnetic field axis and their corresponding u values. The main problem related to this model is that many hysterons are required to reliably predict the hysteresis loops ($N_{hyst} \simeq 1.3 \cdot 10^6$). Equation (4) allows instead to reduce the parameters of the model to only three (σ_H, σ_u, u_0). Another important advantage is that, using proper algorithms, there is the possibility to identify the model parameters with a reduced number of measurements (just a limited amount of hysteresis loops is required). In the work, in particular, the authors use four cycles of sinusoidal hysteresis for the identification of the PM. The error function $f = \frac{1}{3}(f_1^2 + f_2^2 + f_3^2)$ which must be minimized consists of three contributions:

1. MSE error, normalized sample by sample, is evaluated considering the measured and calculated B values of the main cycle (n_l), with $n_s = 500$,

$$f_1 = \frac{1}{n_s} \sum_{k=1}^{n_s} \left(\frac{B_m^{n_l}(k) - B_c^{n_l}(k)}{B_m^{n_l}(k)} \right)^2 \tag{5}$$

   where $B_{m,c}^j(k)$ is the k^{th} element of the j^{th} sequence of the measured and evaluated B fields.
2. The second term is the normalized MSE of the error between the measured value maximum and calculated value of the magnetic induction in the vertices of the cycles:

$$f_2 = \frac{1}{n_l} \sum_{k=1}^{n_l} \left(\frac{\max(B_m^k) - \max(B_c^k)}{\max(B_m^k)} \right)^2 \tag{6}$$

3. The last term is introduced to improve the accuracy of the model by introducing cycle areas for the calculation of hysteresis losses

$$f_3 = \frac{1}{n_l} \sum_{k=1}^{n_l} \left(\frac{A_m^k - A_c^k}{A_m^k} \right)^2 \tag{7}$$

   with $A_{m,c}^j$ areas of the j^{th} calculated and measured hysteresis cycles.

The calculation of the distribution of hysterons can have, in general, more solutions and consists of two stages. Many hours are requested for the first one since lots of parameter combinations need to evaluated with a cost function. Using the previously defined function, the probability of obtaining a local minimum cost function is significantly reduced, thus accelerating the next step that involves the optimization algorithm. The identification procedure is quite expensive from the computational point of view, especially if N_{hyst} is great; however, it must be considered for each material. In Figure 4, the family of the 20 normalized FORCs considered for the training of NN is reported.

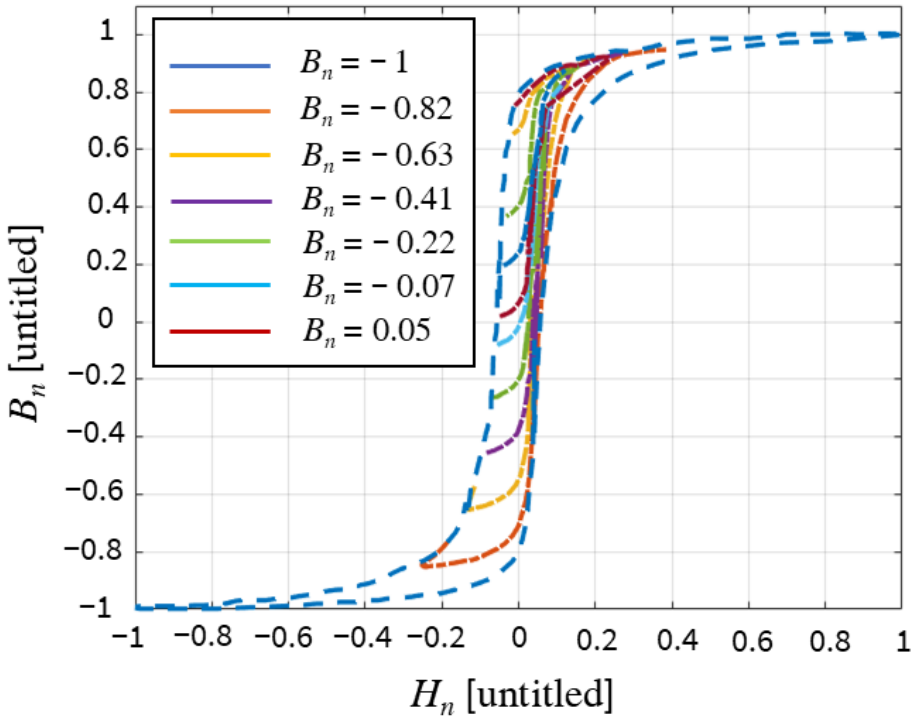


Figure 4. FORCs considered to train the NN. B_n is the normalized magnetic induction.

The hysteresis model based on NN is divided into two steps: firstly, the development of a standard FNN able to reproduce the natural memory of magnetic hysteresis but characterized by a limit in the reproduction of hysteresis loops. Secondly, the usage of specific approaches to stabilize the magnetic accommodation, with the consequent advantage of accurately reproducing the sub-loops. In this way, the resulting model gains considerably more generality, with the possibility to be applied for a large variety of cases. Basically, the model is built as an FNN (structured with two hidden layers each containing 7 neurons (see Figure 5)) which takes in as input the magnetic field and the magnetic induction at the instant $k-1$ and provides as output the differential permeability at the k instant. A sigmoid activation function (or hyperbolic tangent) is applied to the hidden layers' neurons, and a linear transfer function is applied to the output layer. The hyperparameters are experimentally optimized. From the diagram, it is possible to see that the insertion properties of past data are included in the last step; this process is not intrinsically included in the FNN. The value of the magnetic induction at the k instant is evaluated as $B(k) = dB(k) + B(k-1)$. As stated in the previous lines, the FNN provides the value of the differential permeability at the k instant ($\mu(k) = dB(k)/dH(k)$). The differential increment of the magnetic field ($dH(k)$) can be easily obtained using the previous and the actual value of the magnetic field. Finally, $B(k) = \mu(k)dH(k) + B(k-1)$. Weights (70), neuron biases (15), and the training set (the family of 20 Figure 4) are the only elements influencing the net, which is identified only once for each material. The maximum number of epochs considered is 15×10^3, while the performance evaluation is based on MSE. The network is trained six times, and its robustness is tested, each time simulating 20 FORCs, derived as under sampling (with a factor 8) those applied for training. This procedure is adopted to avoid the local minima of MSE, which could be found as a consequence of the training set identification. The best NN obtains an $MSE = 1.88 \times 10^{-3}$. The entire procedure is completed within 30 min.

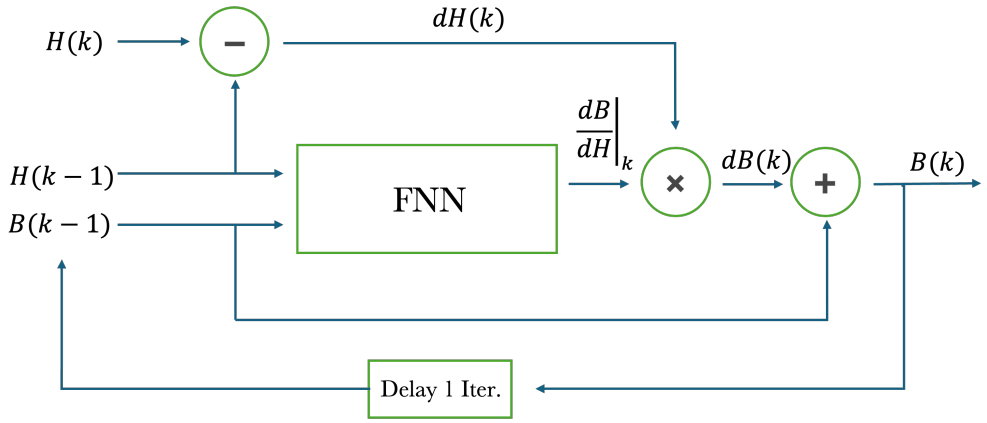
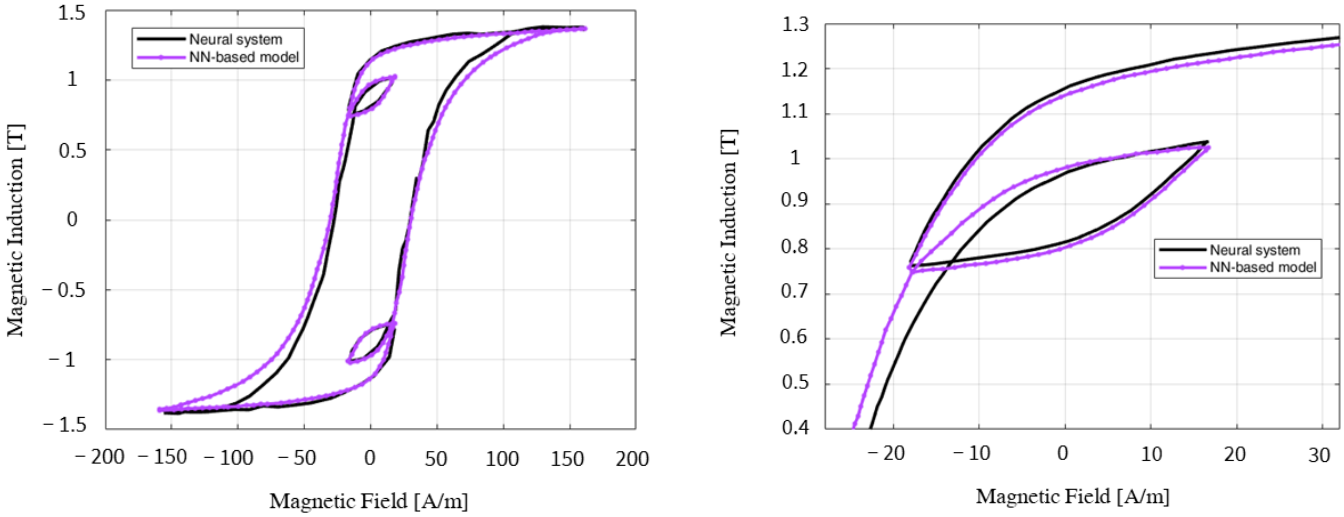


Figure 5. Final NN-based hysteresis model.

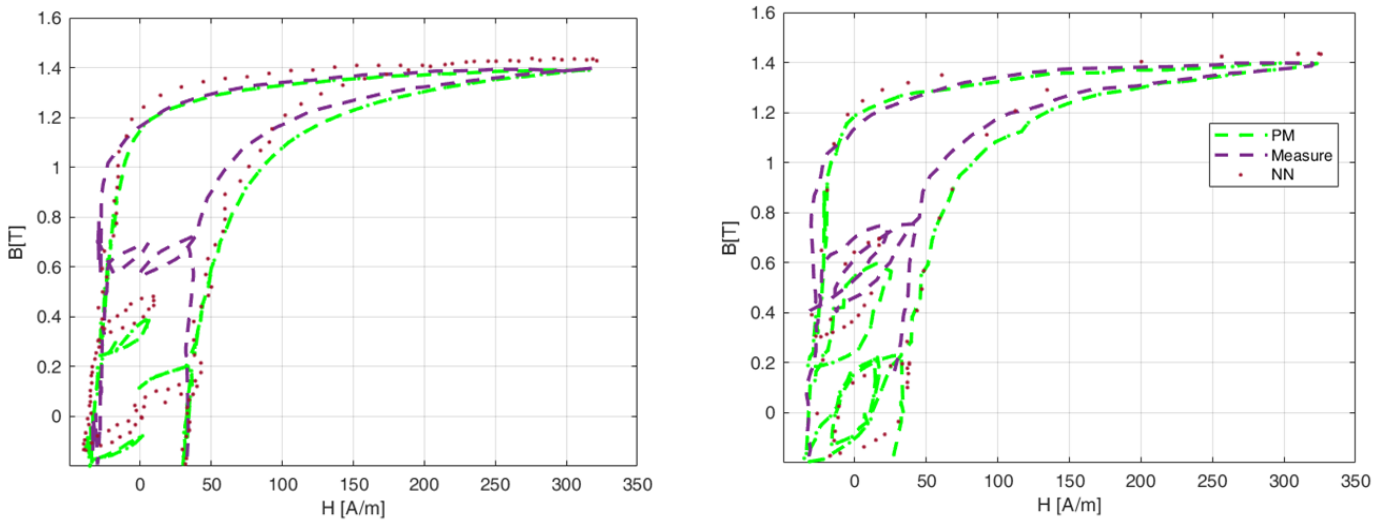
If the input CM is characterized by oscillations between H_1 and H_2 (two chosen ends), and the neural network starts from an arbitrary magnetized state, several periods are necessary to reach a stable magnetization cycle. This feature is called 'accommodation', and it is typical of just some kinds of materials. For example, it does not affect electrical steel, where sub-loops are practically stable and a simple neural system is sufficient, but where present, it requires a high numerical cost. The network used offers a considerable computational advantage thanks to the previously mentioned transplantation technique. Ultimately, the PM is used to create a FORC dataset to be used as a dataset for the neural network, and it has been proven that the implemented neural model is able to replicate the behavior of the PM asking for lower computation effort and reduced memory storage request.

Figure 6 shows the predictions of hysteresis loops, including short loops evaluated with the transplantation technique, comparing a NN-based model and a neural system alone. Different sets of experiment data are used for the simulations. Computations and experiments are shown in Figure 7, considering the hysteresis cycles of one of the analyzed sets. Figure 7a,b show hysteresis loops evaluated imposing a B made by a sinusoidal fundamental waveform, with the superposition of a fifth harmonic. The presence of multiple harmonics in the B waveform generates sub-loops in the resulting hysteresis loop. In particular, a multiple harmonic starts to cause sub-loops when its amplitude reaches a certain fraction of the fundamental harmonic amplitude. A harmonic ratio (hr) coefficient is defined to quantify this aspect, and as a generic criterion, sub-loops start to be visible when $hr = 0.4$. Obviously, the higher hr is, the more evident the sub-loops are in the hysteresis loop. Figure 7a shows the hysteresis loop when $hr > 0.5$, and Figure 7b shows that when $hr = 0.75$. Consistent with what is written in the previous lines, sub-loops are bigger in Figure 7b. Relative to this comparison, it is important point out that the NN-based model performs worse than PM. Indeed, the maximum percentage error from the experimental reference is 18% for the NN model and 11% for the PM. The origin of this inaccuracy is mainly related to a little phase error between the simulated $B(t)$ and the applied input magnetic field $H(t)$. In fact, a phase change in the waveform of the magnetic field is enough to match the experimental cycle. The phase error is always under 10 degrees, and it can be associated (at least partially) with the accuracy in the measurement of the magnetic field. As it can be seen, an error about the positioning of the sub-loops is committed, even if the resulting area is quite similar. This is because the power losses due to the hysteresis evaluated with the two models (NN and PM) agree in a quite good way with the measurements. This result, together with the computational advantages, still makes the method worth considering.



(**a**) (**b**)

Figure 6. Predictions of hysteresis loops, including short loops evaluated with the transplantation technique. Comparison of the NN-based model and neural system alone in the prediction of the hysteresis loop under non-sinusoidal excitation. (**a**) Comparison of the NN-based model and neural system alone; (**b**) sub-loop.



(**a**) Hysteresis loop when $hr > 0.5$. (**b**) Hysteresis loop when $hr > 0.75$.

Figure 7. Comparison of PM and NN-based models with experimental data in the prediction of the hysteresis loop. The NN-based model is characterized by a higher max percentage error with experimental data compared with PM (18% against 11%).

Moreover, it should be noted that since the FNN provides as output the relative differential permeability, the NN-based model can also work in a reversed way (taking B as input and giving H as output), which is exactly what FEM solvers do.

4. Further Neural Architectures

As highlighted in the introductory section, the NN can be varied and in turn combined, depending on the proposed objectives. Within the four macro-areas of action (classification, regression, clustering, and anomaly detection), it is possible to have supervised or unsupervised input and predictive or descriptive logic. Within these, the choices of the hyperparameters, hidden layers, activation functions, or metrics to calculate the error give

rise to infinite possibilities, not least the additions of reinforcement, self-supervised, or contrastive learning techniques. The DL can also work in combination with the ML or, as seen in the previous sections, with known numerical methods. It is clear that they are compatible with PM, extended PM, or other models of magnetic hysteresis. It is therefore not possible to give an exhaustive framework of the possibilities, but the following is limited to listing some papers belonging to some of the main categories already mentioned.

4.1. Recurrent NN, Diagonal RNN, and LSTM

Another important family of neural networks is that of the aforementioned recurring neural networks [54]. They are designed to work with sequential or temporal data, as they contain loops that allow them to maintain information about past events. They are used when data have a temporal or sequential structure, as in the case of time series. Among these, of particular interest are the Long Short-Term Memory (LSTM) (see [55] and Refs. therein) and the Gated Recurrent Unit (GRU) [56]. These types of architectures are designed to better manage long-term dependencies in sequential data than traditional RNNs. They are often used in applications where it is necessary to capture long-term dependencies, such as modeling complex time series. The "diagonal RNN" (dRNN) [32] is a specific type of RNN also used for temporal sequences but, unlike the one in which each recurring unit receives in input its previous outputs and the outputs of the other recurring units in the same temporal passage, dRNN introduces a particular structure in which each recurring unit receives only in input its previous outputs and not the outputs of the other recurring units. This means that recurring connections across time are limited to a diagonal of the connection matrix instead of involving all units. This design simplifies the network structure, reducing the number of connections needed and improving the computational efficiency. They are particularly useful when the long-term relationships between distant time positions are not as relevant as short-term relationships.

An example of such a network's application to the hysteresis process is given by [32]. Here, too, PM is used, but the authors demonstrate that the rate-independent (RI) PM is in fact a dRNN in which the activation function is a binary step. The black box technique is not used, but the used activation function is a manipulated tanh. It is also shown that dRNN is also a versatile rate-dependent (RD) hysteresis system under detailed conditions. Relationships are established through the direction, shape, symmetry and rate dependency of hysteresis cycles and dRNN parameters so that the former can be interpreted through the latter. dRNN formulated in this way can also model RD hysteresis, which is more precise than simple PM because no additional parameters and changes are introduced to the classic dRNN. The model is trained using experimental data of materials with hysterical behavior, and the accuracy is assessed by comparing model predictions with experimental measurements. The training time can be reduced considerably, and machine learning frameworks such as PyTorch can be used. Moreover, the method is general, unlike the various adaptations to the various types of hysteresis that are specific to each experiment. The classic NNs are used for both RI and RD hysteresis models but generally include a single activation function (as we saw in the previous sections, e.g., sigmoid, tanh, and Gaussian), while hysteresis is a multi-valued phenomenon. The authors, through the use of dRNNs, avoid the common use of the enlargement technique of the input space (e.g., in FNN) to expand the action of the model (incorporating, for example, Preisach-type hysterons or other coupling variables, for example, of the historical type), and do not use RBF-type functions with multiple inputs (common to capture non-linear relationships between historical and current variables). This approach reduces computational complexity and gives to the network the ability to learn time dynamics on its own. The dRNN architecture is more effective while maintaining the possibility to model non-linear behaviors such as magnetic hysteresis.

In [33], a Preisach–RNN model is provided to forecast the dynamic hysteresis in ARMCO pure iron, the fundamental soft magnetic material used in particle accelerator magnets, without requiring prior knowledge of the material and its microstructural be-

havior. The dynamic aspect includes the dependence of the hysteresis cycle on the rate of change due to the interaction between electric and magnetic fields. A novel validation method is suggested to identify the model's parameters through a RNN coupled with Preisach play operators. In general, a RNN consists, like the aforementioned NN, of an input layer, a hidden layer, and an output layer. The input and output layers involve feed-forward connections, and the hidden layer, recurrent ones. The input vector named $v(t)$ is processed at the input layer at each time step t. Later, $v(t)$ is summed to the bias vector b_1 and multiplied for w_1, the input weight matrix. Equally, the internal state $z(t)$, slowed down by a number of time instants d, is multiplied by the gain factor w_h and added to the input state according to the formula

$$z(t) = f_h[w_1 \cdot v(t) + b_1 + w_h(z(t-d))] \tag{8}$$

with the $f_h(x)$ activation function (here tanh). The internal state $z(t)$ is then added with bias b_2, multiplied by the weight w_2, and the result is passed through a linear activation function $f_0(x)$ as follows:

$$y(t) = f_0[w_2 \cdot z(t) + b_2]. \tag{9}$$

where $y(t)$ is the predicted output at time t. The layers scheme can be seen in Figure 8. Here, too, the Levenberg–Marquardt algorithm is used as an optimization model for the output layer. It is a non-linear least squares optimization algorithm incorporated into the backpropagation algorithm for training NN [57]. If $\overline{y}(t)$ is the real datum, the algorithm aims to optimize the output through the formula

$$g(t) = \frac{1}{2}(\overline{y}(t) - y(t))^T(\overline{y}(t) - y(t)). \tag{10}$$

It leads to updating the weights according to the following

$$w_k(t+1) = w_k(t) + \eta\left(-\frac{\partial g}{\partial t}\right), \tag{11}$$

with $\eta \in \mathbb{Z}^+$ representing the learning rate.

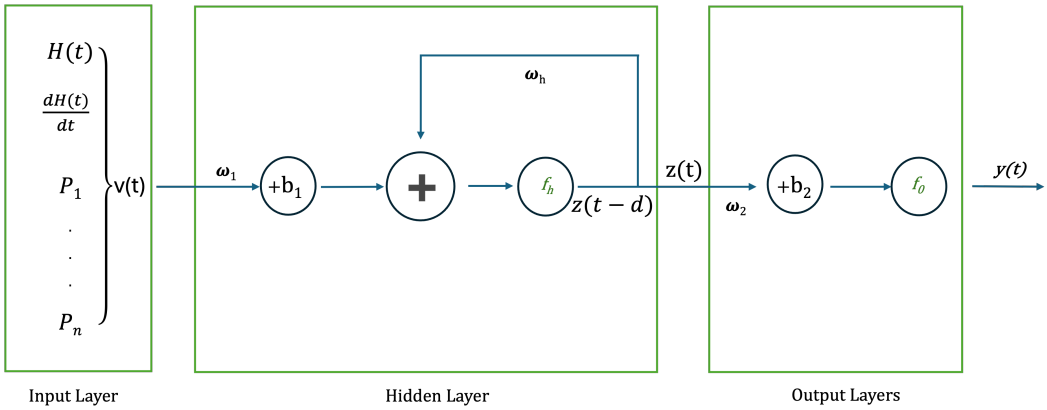


Figure 8. PM-RNN scheme.

By training with only six different hysteresis loops at three frequencies, the proposed model is able to predict the magnetic flux density of ARMCO pure iron with a NRMSE better than 0.7% and can predict dynamic behavior for both the main and sub-cycles. The model's accuracy in predicting data that have not been measured is demonstrated through its evaluation using ramp rates that are not utilized in the training procedure. In the field of materials science, the Preisach model, based on a RNN, is shown to accurately describe ferromagnetic dynamic hysteresis also when trained with a limited amount of data.

In [38], the authors exhibit the ability of an LSTM network to capture the intricate hysteretic dynamics of piezoelectric actuators (PEAs). The network is established to represent the sophisticated motion of PEAs, which incorporates static hysteresis or high-order dynamics. By using datasets of input/output pairs obtained experimentally, excitations of various frequencies and amplitudes, the network is trained and evaluated. The preliminary findings indicate that the LSTM network can provide adequate precision in a wide frequency range, even for the simplest topology, such as a single layer with one cell. Thus, LSTM networks may offer a novel approach to approximate the dynamics also in complex engineering systems (see [55] for more details on the LSTM scheme and mathematical model).

In [58], ordinary differential equations are employed by the authors to model and quantify hysteresis, which is manifested in sequentiality and historical dependence. They propose a neural oscillator, "HystRNN", which is inspired by coupled oscillatory RNN and phenomenological hysteresis models to update the hidden states (HSs). The performance is measured for the purpose of predicting generalized scenarios, which involve first-order reversal curves and minor loops. The results exhibit the capability of HystRNN to generalize its performance to untrained parts (essential feature for hysteresis models), as it has been discussed extensively. The paper draws attention to the advantage of neural oscillators over the common RNN-based methods in detecting complex hysteresis patterns in magnetic materials, where traditional RD methods are not sufficiently efficient to catch intrinsic non-linearities. The methodology uses a structure similar to that of RNN with a difference included in the HS upgrade. Indeed, HystRNN applies ODEs for updating them. The procedure engages two inputs, H and B_{-1}, which are mapped to B. The modeling process collects a number n_e of experimental data points and the number of training points is $N = n_e - 1$. The technique shares some analogies with the FNN architectures used for modeling hysteresis but differs by including a recurrent affinity capturing longer-time dynamics and output dependencies. The cited ODE is a second-order ODE

$$y'' = f_1\Big(w_1 y + w_1^1 y' + v_1 u + b_1\Big) + f_2\Big(w_2 \mid y \mid^2 + w_2^2 \mid y' \mid^2 + v_2 \mid u \mid^2 + b_2\Big), \quad (12)$$

with $y = y(t) \in \mathbb{R}^m$ denoting the HS of the HystRNN, y' the time derivative, y'' the second-order time derivative, and $w_{1,2}, w_{1,2}^{1,2} \in \mathbb{R}^{m\times m}$ and $v_{1,2} \in \mathbb{R}^{m\times n}$ the weights matrices. $n = N \times 2$. The aggregated training data correspond to the time t. $u = u(t) \in \mathbb{R}^n$ is the input, $b_{1,2} \in \mathbb{R}^m$ the bias vector, and $f_{1,2} = \tanh(u)$ the activation functions. The authors here introduce a reduction of the differential order by using the auxiliary variable $z = y'(t) \in \mathbb{R}^m$ and obtaining the first-order system

$$\begin{cases} y' = z \\ z' = f_1(w_1 y + w_1^1 z + v_1 u + b_1) + f_2(w_2 \mid y \mid^2 + w_2^2 \mid z \mid^2 + v_2 \mid u \mid^2 + b_2) \end{cases}. \quad (13)$$

They then use an explicit scheme to discretize the system for $0 < \Delta t < 1$. The output obtained is finally calculated for each recurring unit. The method is evaluated using four metrics: (1) "L2-norm" for the measure of the Euclidean distance between predicted and real values; (2) "explained variance score" denotes prediction accuracy, catching variance proportion; (3) "maximum error" discovers important prediction discrepancies as potential outliers; (4) "mean absolute error" evaluates the mean differences between predictions and real values for general precision. Error types 1, 3, and 4, together with the higher explained variance, imply enhanced performance. The trained architecture is tested in two different eventualities concerning the prediction of two FORCs and two minor loops. For the FORC prediction, two different lengths of sequence (199 and 399) are tested. For minor loops prediction, a sequence with a length of 399 each is used. As with network learning, sequence length tests also depend on the data generated by the PM to evaluate the model. HystRNN is confirmed by predicting first-order reversal curves and minor loops, after training the model only with major loop data. The results emphasize the primacy of HystRNN in ably catching intricate non-linear dynamics, out-performing conventional

RNN architectures such as LSTM or GRU on various metrics. This result is imputable to its faculty to comprehend sequential information, historical dependencies, and hysteretic features, finally reaching generalization competences.

4.2. Convolutional NNs and Temporal CNNs

Convolutional neural networks (CNNs) are an advanced class of artificial neural networks designed for the processing and analysis of structured data in matrix form as images, audio and video. Introduced in [59] in the 1990s, CNN revolutionized the field of artificial vision by their ability to automatically learn relevant features from raw data. They consist of a convolutional layer, the heart of the architecture, composed of several layers, in which a series of filters (or kernels) is applied to the input images, generating feature maps [55,60]. These filters are able to capture various local features such as edges, corners and textures, essential for understanding the image. After each convolution layer, a pooling layer (usually max pooling) reduces the spatial dimension of the feature maps, keeping the most important information and reducing the computational complexity. This process makes the model more robust under variations and translations of the image. Finally, fully connected layers are composed in the same way as those in traditional NN. These layers combine the characteristics learned during the convolutions to carry out the final classification or other recognition activities [55]. They have been successfully used in a wide range of applications, including image recognition or semantic segmentation. CNNs learn to extract features at different levels of abstraction, from edge detection to complex shapes, and share weights, drastically reducing the number of parameters to learn and improving the computational efficiency but requiring large amounts of labeled data for effective training. The training also requires a considerable computational cost or specialized hardware such as a GPU. The high flexibility of the CNN can be adapted for the analysis of multi-dimensional data, such as magnetic hysteresis data.

In [35], the authors analyze the temperature variation modifying the magnetic behavior of ferromagnetic cores, which can have impact on the performance of electrical devices. To build a temperature-dependent hysteresis model to accurately calculate the electromagnetic features, in this case , it can have a significant impact. A temporal convolutional network (TCN) in combination with the play operator method is developed in the paper. To introduce the temperature effect, the suggested model uses the temperature-dependent spontaneous magnetization intensity as an input. The classical play model is history dependent and rate independent and designed for static magnetic hysteresis calculations. It can be represented by Equation (14), in which a series of operators is integrated under the action of a rate-independent shape function f_z

$$H = P(B) = \int_0^{B_s} f_z(p_z(B))dz \tag{14}$$

where B_s is the saturation magnetic flux density, and p_z is the play operator expressed as follows:

$$p_z(B) = \max\Big(\min\Big(p_z^0, B + z\Big), B - z\Big) \tag{15}$$

with p_z^0 as the value of the previous moment. Such a rate-independent model struggles to predict the dynamic loss accurately. However, the history-dependent play model integrated with TCN is transformed into a dynamic, rate-dependent magnetic hysteresis model. The performance indicator employed to quantify the error between the model and experimental measurements is the Normalized Root Mean Square Error (NRMSE). The process of model validation involves selecting the suitable model hyperparameters and making sure that the model is robust to new data. The Bayesian optimization algorithm is employed to optimize the hyperparameters and improve the accuracy of the model training outcomes. The results exhibit that the provided model can accurately forecast the hysteresis features of materials, both under varying temperature and frequency conditions.

In [36], the problem of how the output force of the pneumatic blow-off actuators is critical for their applications is addressed. Its force control poses a great challenge due to the strong asymmetric hysteresis posed by its material's hyperelasticity and air's high compressibility. The author proposes a hybrid model named the CNN-AUPI (amplitude-dependent unparallel Prandtl–Ishlinskii (PI))-based force–position hysteresis modeling method for soft actuators. AUPI is a modified model built on the traditional PI, which is a weighted superposition of the multiple play operators as illustrated in the following:

$$F_{k_i}(u(t)) = \max\{u(t) - k_i,\ \min\{u(t) + k_i,\ F_{k_i}(u(t^-))\}\}, \quad t^- = t + 0^+. \tag{16}$$

A single play operator is a time-varying equation. The PI superimposed by multiple play operators can only fit symmetric hysteresis curves because of the symmetry of the play operator. Built on PI, a UPI (Unparallel PI) is considered to model asymmetric hysteresis. The mathematical formulation is

$$F_{k_i,a_i}(u(t)) = \max\{u(t) - k_i,\ \min\{a_i(u(t) + k_i),\ F_{k_i,a_i}(u(t^-))\}\}, \quad t^- = t + 0^+ \tag{17}$$

describing the asymmetric event of the hysteresis loop by multiplying a factor a_i on the falling edge of the play operator. The UPI model for the soft joint actuator accurately depicts the asymmetric hysteretic behavior at a specific inflation pressure but develops into being unreliable at varied inflation pressures. It is incorporated such that the soft joint actuator's maximum rotation angle "A" is introduced at each inflation pressure [61] to assure that the model is able to forecast hysteresis at diverse air pressures. The AUPI model includes various weighted UPI operators and the constant term

$$y = \left(\sum_{l=0}^{N_r} w_l F_{k_l,a_l}(u(t)) \right) + b, \tag{18}$$

where y is the AUPI model output and b is a constant to be identified, introduced to represent the actuator hysteresis curve features at the starting point. The number of UPI operators, experimentally set, is N_r, and w_i is the density coefficient of the UPI operators. w_i is included to describe a segmented form involving the loading and unloading process of the hysteresis curve, respectively. Furthermore, two non-linear functions frequently used in hysteresis modeling, as density coefficient functions, are applied to AUPI. Although the AUPI model has the capability to represent the asymmetric hysteresis event and generalize the results, its fitting accuracy is not strong enough under the inflation pressure independent of the training data. For this aim and to further improve the accuracy and generalization model capability, a CNN is matched. The CNN mechanism obtains the general characteristics of hysteresis information, avoids overfitting, and dramatically enhances the composite model's accuracy and generalization ability. The ReLU activation function provides advantages such as fast convergence and a lack of gradient saturation or disappearance (with respect to common sigmoid and tanh functions). Since the convolution layer only draws the links between local characteristic nodes, the combination of the CNN with an AUPI model obtains the general relationships between the individual nodes of the feature map. The MAE, MSE, maximum relative error, mean output force error, and R-square quality of fit are used to quantitatively describe the advantages of the models. Experimentally, it is shown that the CNN-AUPI model has brilliant hysteresis fitting for soft joint actuators, with a maximum relative error of only 6.1% and a quality of fit of more than 0.99. Other hysteresis models such as classical PI and improved PI have been compared, and the results show that the CNN-AUPI mode has a strong modeling accuracy and high prediction ability, thus providing an encouraging method for soft actuator hysteresis modeling. Furthermore, it is generalizable and suitable to model asymmetric hysteresis for different kinds of soft actuators.

4.3. Generative Adversarial Networks (GANs)

Generative neural networks, e.g., generative adversarial networks (GANs), are used to generate new synthetic data which follow the same distribution of input data. They are among the most intriguing ideas in computer science today. An adversarial process is employed to train two models simultaneously. A generator ("the artist") learns to create images that look real, while a discriminator ("the art critic") learns to tell real images apart from fakes [37]. As training progresses, the generator improves its ability to create images that appear real, while the discriminator improves its ability to distinguish them. The process reaches equilibrium when the discriminator can no longer distinguish real images from fakes. This process trains the GAN to generate real images that may not be found in the original dataset. This approach can be useful to generate additional data to train NN models or to explore the hidden features of magnetic hysteresis data.

In [62], GAN is discussed as a tool to predict magnetic field values at random points in space using point measurements. Obtaining high-resolution magnetic field measurements may be difficult or impractical, which is why this technique is particularly useful for scientific and real-world applications. The implemented GAN consists of two main neural networks: a generator, which predicts missing magnetic field values, and a critic, which calculates the statistical distance between real magnetic field distributions and the generated field. The architecture is shown in Figure 9 and consists of an input, 3D magnetic field (measured in a 2D rectangular area), and an output (an inter- or extrapolated 3D magnetic field in this region). The fields are multiplied for a binary mask m during the training, according to the following (we recall that $\odot$ is traditionally the symbol for the Hadamard product):

$$B_{sp} = B \odot (1 - m). \tag{19}$$

The two-step generating process is designed in the style of residual learning [63]:

$$B_c = G_c(B_{sp}, m), \qquad \tilde{B} = G_f(B_c \odot m + B_{sp}, m). \tag{20}$$

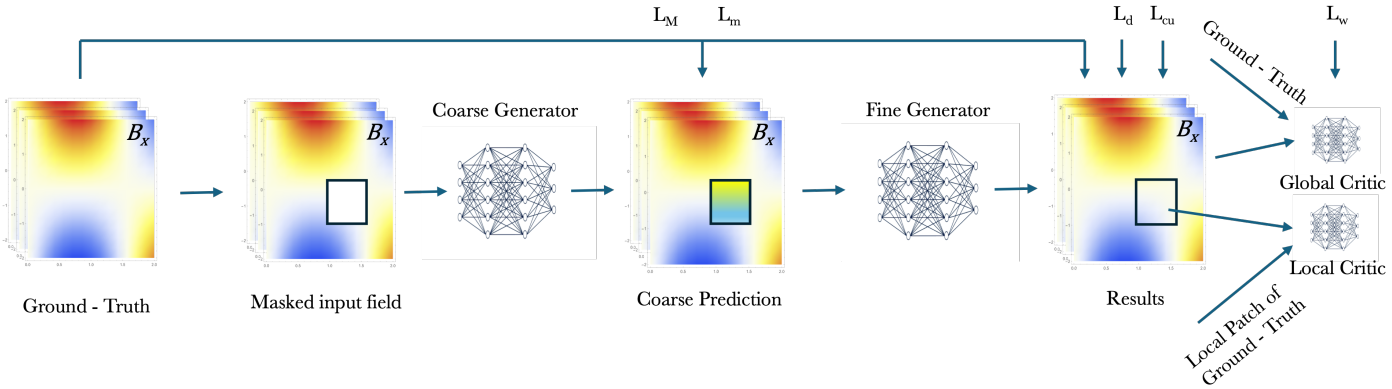


Figure 9. GAN architecture. A two-step generation process with down/up-sampling across multiple convolutional layers. It provides missing field values of a masked input magnetic field. The results, calculated by local and global critic, consist of several convolutional layers. Error functions $L_{\{\dots\}}$ are evaluated for updating the parameters of the generator networks in order to minimize the general loss function.

The generator network G_c generates a coarse prediction by applying a sequence of convolutional layers on B_{sp} and the applied mask m. At the start, the input field is reduced to a lower resolution with an increased number of channels. This approach ensures that the same quantity of information is preserved while making successive convolutions computationally less intensive. Later, to enhance the model's field of view and enable encoding at multiple scales, many convolutions with different scaled filters are executed

on the down-sampled image. Lastly, the data are up-sampled with interpolations to the original size, providing a coarse forecasting B_c. Another generator network G_f takes B_c and B_{sp} as input and provides $\tilde{B}$ in similar way to G_c. Along with that, the magnetic field is split up into small patches of 3×3 pixels in a second branch. The reconstruction is improved by calculating the corresponding importance between these patches and missing pixels. The purpose of this kind of contextual attention is to overthrow localization in the convolutional layers and enhance it with a comprehensive flow of information from field pixels that are magnetically distant. The convolution and attention branch are linked before up-sampling to the original resolution. On $\tilde{B}$, the losses L_M, L_m , L_d, and L_{cu} can directly be evaluated. For the adversarial loss L_w, we necessitate the employment of a critic NN. It needs to split the critic into a global critic network to evaluate the whole image, and a local critic network to determine the quality of the filled-in regions. The model can predict missing magnetic field values by training the generator to minimize this statistical distance, as well as minimizing reconstruction and physical losses based on Maxwell's equations. The average reconstruction error is 5.14% with a consistent region of field points and 5.86% when only a few spot measurements are available. These results prove that the technique can be used effectively to reconstruct missing magnetic fields and could have applications in various fields requiring the measurement and the analysis of magnetic fields. This novel method is not able to perform painting taskswhere large parts of the magnetic field, calculated relative to the general measurement area, are missing. Moreover, the physics-informed learning-based method produces better performance results when compared to the other common methods. Furthermore, when regions (16×16 pixels) of measurements are given (instead of only (1×1 pixel)), the Gaussian processes outperform the procedure but with the inference time of magnetic field forecasting being two orders of magnitude higher. In certain applications, such as the simultaneous mapping and localization performed in robotics, the paradigm could be a compromise between accuracy and computational time. The authors suggest that it would be very attractive to make use of the fact that closed Poisson problems could be solved from the boundary values around the missing field information. Hence, the generator neural network could be trained to predict missing field measurements from only these values in the input layer.

4.4. FNN and Extended PM

As mentioned in the introductory section, various mathematical models and their variants can be exploited to reproduce the hysterical behavior of the ferromagnetic materials to be considered. Artificial neural networks, as the usual NNs, involving time-delay, multi-layer perceptron, and RNN, sometimes are inadequate for learning entirely hysteretic behaviors. Appropriate memory to deal with hysteresis as a non-unique non-linear event [64] is lacking. Based on PM and PI hysteresis models, built on hysteresis operators relay, play and stop, in [65], the authors suggest a novel NN, the Prandtl neural network (PINN). It provides only one hidden layer with stop neurons. It is a linear combination of many stop operators as in the PI model. The model can be applicable to the hysteresis following Masing rules [66]. In [67], the same authors extend PINN by inserting an extra hidden layer with sigmoidal neurons to non-Masing hysteresis based on the Preisach neural network (PMNN). Here, the stop neurons in the first hidden layer are mapped into the output layer through a non-linear mapping by the second hidden layer, like in the PM. Further extension provides a novel hysteresis operator by putting together stop and play operators and using it in a NN called the generalized Prandtl neural network (GPINN), by which non-congruent hysteresis comportment can be mocked. Both PMNN and GPINN are diverse extensions of PNN. In [34], these extensions are joined into a new NN called the Extended Preisach Neural Network (EPNN). Furthermore, it is improved for the RD hystereses that the previous extensions lack. It includes one input layer, one output layer and two hidden layers. The input and output layers consist of linear neurons; the first hidden layer, different from the PMNN, involves Normalized Decaying Stop (NDS) neurons, whose activation mechanism is constructed after the decaying stop processor

with a unit threshold ($r = 1$). This kind of operator can generate non-congruent hysteresis loops. In the input layer, it includes $x(t)$ input data and $\dot{x}(t)$, the rate at which $x(t)$ changes, in order to provide to EPNN the ability of learning RD hysteresis loops. In the second hidden layer, sigmoidal neurons are included. They help the NN learn non-Masing and asymmetric hysteresis loops very smoothly. The envisaged technique allows the simulation of both RI and RD hystereses with either congruent or non-congruent loops and symmetric or asymmetric loops. For the EPNN training, a novel hybridized algorithm is adopted, built on a combination of GA and the optimization method of sub-gradient with space dilatation. By applying the proposed model to different hysteresis processes, from various engineering areas, with different features, the generality of the model is evaluated. Results indicate the success of the model in the identification of the examined hysteresis and the arrangement with experimental data.

4.5. Deep Operator Networks

Deep Operator Networks (DeepONets) [68] are a type of neural network designed to learn non-linear operators, mapping from one functional space to another. This makes them particularly suitable to model the constitutive laws governing complex phenomena such as magnetic hysteresis. In a DeepONet, the input is not a single vector as in a traditional FNN but a function. The architecture is designed to manage these functions as inputs, thus allowing to directly learn the non-linear relationships between entire functional spaces. The architecture includes three main sections: the Branch Network, which acquires a function as the input and extracts its characteristics; the Trunk Network, which captures the coordinates in the target space and transforms them into a representation that can be combined with the output of the Branch Network; and the Combining Mechanism, in which the output of the two networks is combined to produce the final map between the input function and the target. The structure follows a mathematical model reported in the following Theorem [68].

Theorem 1 (Universal Approximation Theorem for Operator)**.** *Suppose that σ is a continuous non-polynomial function, X is a Banach space, $K_1 \subset X$ and $K_2 \subset \mathbb{R}^d$ are two compact sets in X and $\mathbb{R}^d$, respectively, V is a compact set in $C(K_1)$, and G is a non-linear continuous operator, which maps V into $C(K_2)$. Then, for any $\varepsilon > 0$, there exist constants $c_i^k, \xi_{i,j}^k, \theta_i^k, \zeta_k \in \mathbb{R}$, $x_j \in K_1$ with $i = 1, \dot{n}, k = 1, \dot{p}$, $j = 1, \dot{m}$, $n, p, m \in \mathbb{Z}^+$ such that*

$$\left| G(u)(y) - \sum_{k=1}^{p} \underbrace{\sum_{i=1}^{n} c_i^k \sigma\left(\sum_{j=1}^{m} \xi_{i,j}^k u(x_j) + \theta_i^k \right)}_{Branch} \underbrace{\sigma(w_k \cdot y + \zeta_k)}_{Trunk} \right| < \varepsilon, \qquad \forall u \in V, y \in K_2. \quad (21)$$

This approximation theorem indicates the potential application of NN to learn non-linear operators from data, in the same way as ordinary NNs, where we learn functions from data but do not obtain information on how to learn efficiently. The key point is that the new operator G is like a neural network that is able to infer useful information from known and unknown data. The general accuracy of NNs can be described by separating the global error into three main types: approximation, optimization, and generalization errors (see [68] and the Refs. therein). But Theorem 1 guarantees a small approximation error for an adequately significant network, and it does not consider the critical optimization and generalization errors at all, which are often preponderant contributions to the total error in effect. Useful NNs should be simple to train, which means to exhibit small optimization errors and generalize well to unknown data (namely with irrelevant generalization errors). To prove the ability and efficacy of learning non-linear operators by NN, the problem is considered as generally as possible by using the weakest acceptable restriction on the sensors and training dataset. The DeepONet structure makes it possible to achieve small total error and model complex processes involving whole functions, overcoming the limitations of

traditional neural networks when it comes to generalizing new inputs. Applications are in several fields, including the modeling of magnetic hysteresis.

The limits of traditional neural architectures, including RNN and appropriate variants (Gru, LSTM, and so on), occur from their capability to learn only fixed-dimensional mappings between magnetic fields. Such networks cannot model mappings between functions in continuous domains [69,70]. In [39], the authors propose neural operators (NOs) to model the hysteresis relationship between magnetic fields to deal with these challenges. Common NNs learn fixed-dimensional mappings, while NOs approximate the underlying operator, building a mapping between H and B fields, to predict the material responses (B fields) for new H fields. Precisely, NOs can approximate continuum mappings even when used on discrete data, permitting them to generalize to new H fields. More particularly, two notable neural operators, DeepONet and Fourier NO (see Figures 10 and 11 and [39] for each detail), are employed to predict new first-order reversal curves and minor loops (new means that they do not belong to the training dataset). Furthermore, an RI Fourier NO is proposed to forecast material responses at sampling rates diverse from those utilized during training to incorporate the RI features of magnetic hysteresis. The numerical experiments presented demonstrate that NO adequately models magnetic hysteresis, overcoming the conventional neural recurrent techniques on different metrics and generalizing to new magnetic fields. The observations underline the benefits of using neural operators for modeling hysteresis under varying magnetic conditions, highlighting their importance in characterizing the magnetic material of devices.

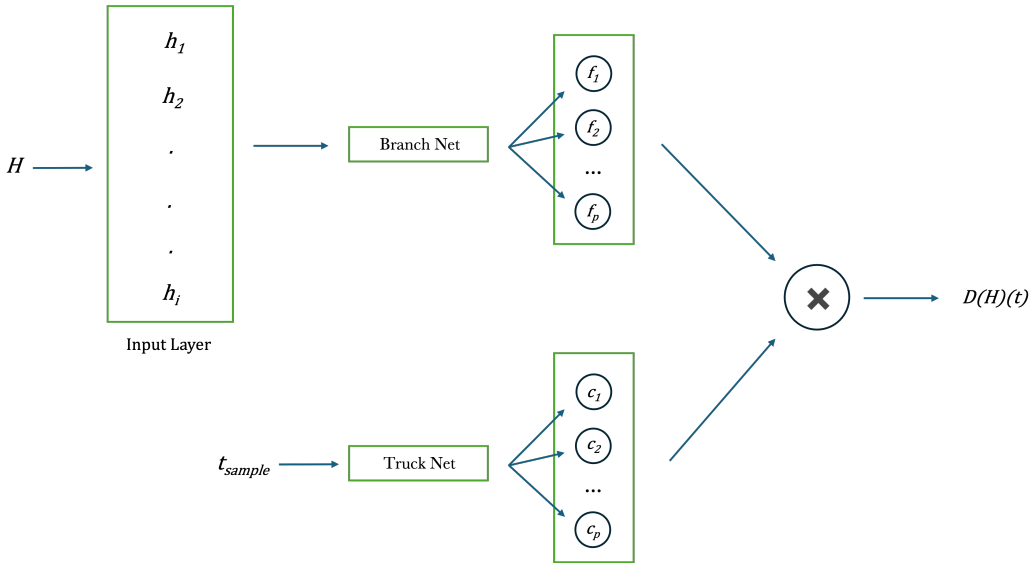


Figure 10. DeepONet architecture composed by 2 different FNNs, branch and trunk net, whose outputs are matched using a dot product to approximate the *B* fields.

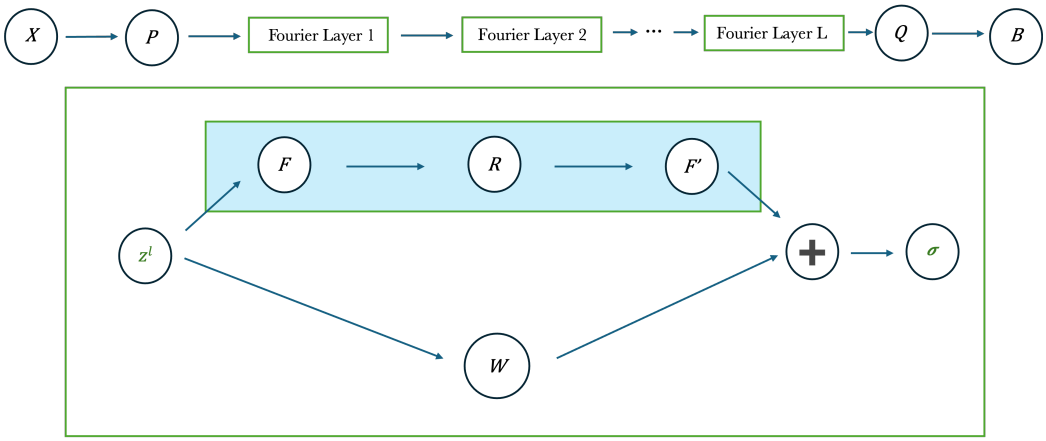


Figure 11. Fourier neural operator architecture (FNO). The input is $X := [h_i, t_{sample}]$ (for RD-FNO it is $X := h_i$). The input is passed through the projection tensor (P) and the Fourier layers, and, at the end, is downscaled (Q) to approximate the B field. In light blue Fourier layers, parameterized by learnable tensors W and R.

5. Conclusions

This paper examines a wide range of neural network approaches to model magnetic hysteresis, pointing out how different architectures can be used effectively to capture and predict the complex behavior of ferromagnetic materials. From the use of convolutional and recurrent neural networks for GAN- or DeepONet-based models, each approach has demonstrated its strengths in addressing specific aspects of the dynamics under consideration. The analysis clearly shows that neural networks, thanks to their deep learning and generalization capabilities, are powerful tools for modeling magnetic hysteresis, regardless of the mathematical model used for the latter. In particular, RNN and CNN models have proved effective at predicting hysteresis dynamics even without the need for prior knowledge of the material details. GAN has shown considerable potential in reconstructing missing magnetic fields. DeepONet generalizes models essential for scenarios where prior training on varying magnetic fields is impractical, among other things. These results not only broaden the understanding of magnetic hysteresis but also offer new perspectives for the practical application of these models in various industrial and scientific fields. The deep learning techniques analyzed, with further optimizations and validations, can lead to significant improvements in the design and management of advanced magnetic devices. Last but not least, although each methodology has specific advantages, it is clear that a combination of different neural network (hybrid models) approaches could offer a more robust and accurate solution for magnetic hysteresis modeling. The integration of deep learning techniques with knowledge of underlying physical phenomena continues to be a promising direction for future research and applications. The possibilities of creating hybrid forms of neural networks are very wide. This overview is a starting point for developing many other neural network architectures, building upon the successes already achieved and avoiding their limitations. These networks must first be developed and tested from a theoretical point of view, and then applied to the various experiments. Only at the end of a comparison between the theoretical and practical results can the actual advantage in choosing one methodology over another be established.

As a concluding remark, the authors are engaged in developing the appropriate neural networks that can boost the technology of the additive manufacturing of soft magnetic components, with particular attention given to the modeling of the magnetization processes and the simulation of the electrical equipment when complex geometries and sophisticated shapes are required for magnetic components. The analysis proposed in the present context has provided additional tools to optimize the prediction of magnetic hysteresis processes and thus be able to intervene in advance in the experimental phase, thereby reducing costs and time considerably. The expected results may thus contribute to a reduction in the

waste materials and energy consumption in the production and lifecycle of the magnetic components involved in electrical machines, actuators, and power converters.

Author Contributions: Conceptualization: S.L., M.L.G., A.S. and A.F.; methodology: S.L.; data curation: S.L., G.A., E.F. and F.V.; validation: S.L., G.A., E.F., F.V., M.L.G., A.S., F.S., V.B., A.D.S. and A.F.; formal analysis: S.L., G.A., E.F. and F.V.; writing—original draft preparation: S.L.; writing—review and editing: S.L., G.A., E.F., F.V., M.L.G., A.S., F.S., V.B., A.D.S. and A.F. All authors have read and agreed to the published version of the manuscript.

Funding: This work is supported under the Project No. 2022ARNLRP funded by the "European Union–Next Generation EU, Mission 4 Component 1 CUP J53D23000670006".

Data Availability Statement: The original contributions presented in the study are included in the article, further inquiries can be directed to the corresponding author.

Acknowledgments: The work of S. Licciardi has been developed in the framework of the project "Network 4 Energy Sustainable Transition—NEST", code PE0000021, CUP B73C22001280006, Spoke 7, funded under the National Recovery and Resilience Plan (NRRP), Mission 4, by the European Union—NextGenerationEU. The work of E. Francomano has been supported by "MUR (Ministero dell'Università e della Ricerca) through the PNRR project ICON-Q, Partenariato Esteso NQSTI-PE00000023, Spoke 2".

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Kittel, C. *Introduction to Solid State Physics*, 8th Ed.; John Wiley & Sons, Inc.: Hoboken, NJ, USA, 2005.
2. Mencuccini, C.; Silvestrini, V. *Fisica II Elettromagnetismo*; Liguori, Ed.; Napoli, Italy, Casa Editrice Ambrosiana, 1988; ISBN 88-207-1633-X.
3. Chitarin, G.; Gnesotto, F. *Elettrotecnica 1—Principi*; Società Editrice Esculapio: Bologna, Italy, 2020.
4. Jiles, D.C.; Atherton, D.L. Theory of ferromagnetic hysteresis. *J. Appl. Phys.*, **1984**, *55*, 2115.
5. Liu R.; Gu C.; Sun J.; Shu F.; Tang B. Analytical Inverse Preisach Model and Its Comparison With Inverse Jiles–Atherton Model in Terms of Accuracy and Computational Speed. *IEEE Trans. Magn.* **2023**, *59*, 7300605.
6. Salvini, A.; Fulginei, F.R. Genetic algorithms and neural networks generalizing the Jiles-Atherton model of static hysteresis for dynamic loops. *IEEE Trans. Magn.* **2002**, *38*, 873–876.
7. Deželak, K.; Petrun, M.; Klopčič, B.; Dolinar, D.; Štumberger, G. Usage of a Simplified and Jiles–Atherton Model When Accounting for the Hysteresis Losses Within a Welding Transformer. *IEEE Trans. Magn.* **2014**, *50*, 7300404.
8. Yang, Y.; Yang, B.; Niu, M. Parameter identification of Jiles–Atherton model for magnetostrictive actuator using hybrid niching coral reefs optimization algorithm. *Sens. Actuators A* **2017**, *261*, 184–195, ISSN 0924-4247.
9. Stoner, E.C.; Wohlfarth, E.P. A mechanism of magnetic hysteresis in heterogeneous alloys. *Philos. Trans. R. Soc. Lond. Ser. A*, **1948**, *240*, *826*.
10. Mayergoyz, I.D. *Mathematical Models of Hysteresis and Their Applications*; Elsevier: Amsterdam, The Netherlands, 2003; ISBN 978-0-12-480873-7.
11. Al Janaideh, M.; Al Saaideh, M.; Tan, X. The Prandtl–Ishlinskii Hysteresis Model: Fundamentals of the Model and Its Inverse Compensator [Lecture Notes]. *IEEE Contr. Syst. Mag.* **2023**, *43*, 66–84.
12. Feng, Y.; Li, Z.; Rakheja, S.; Jiang, H. A Modified Prandtl-Ishlinskii Hysteresis Modeling Method with Load-dependent Delay for Characterizing Magnetostrictive Actuated Systems. *Mech. Sci.* **2018**, *9*, 177–188.
13. Gan, J.; Zhang, X. A review of nonlinear hysteresis modeling and control of piezoelectric actuators. *AIP Adv.* **2019**, *9*, 040702.
14. Preisach, F. Über die magnetische Nachwirkung. *Z. FüR Phys.* **1935**, *94*, 277–302.
15. Mayergoyez, I.D. *Mathematical Models of Hysteresis*; Springer: New York, NY, USA,1991.
16. Sarker, P.C.; Guo, Y.; Lu, H.Y.; Zhu, J.G. A generalized inverse Preisach dynamic hysteresis model of Fe-based amorphous magnetic materials. *J. Magn. Magn. Mater.* **2020**, *514*, 167290, ISSN 0304-8853.
17. Ge, P.; Jouaneh, M. Generalized preisach model for hysteresis nonlinearity of piezoceramic actuators. *Precis. Eng.* **1997**, *20*, 99–111.
18. Mayergoyz, I.D.; Friedman, G. Generalized Preisach model of hysteresis. *IEEE Trans. Magn.* **1988**, *24*, 212–217.
19. Heslop, D.; McIntosh, G.; Dekkers, M.J. Using time-and temperature-dependent Preisach models to investigate the limitations of modelling isothermal remanent magnetization acquisition curves with cumulative log Gaussian functions. *Geoph. J. Intern.* **2004**, *157*, 55–63.
20. Bertotti, G.; Mayergoyz, I.D. *The Science of Hysteresis*; Elsevier: Amsterdam, The Netherlands, 2005; ISBNs 0080540783/9780080540788.
21. Amann, A.; Brokate, M.; McCarthy, S.; Rachinskii, D.; Temnov, G. Characterization of memory states of the Preisach operator with stochastic inputs. *Phys. B Condens. Matter* **2012**, *407*, 1404–1411, ISSN 0921-4526.
22. Visintin, A. *Differential Models of Hysteresis*; Springer: Berlin, Germany, 1994; pp. 10–29.

23. Ma, Y.; Li, Y.; Sun, H.; Yue, S.; Chen, R. Research on the inverse vector hysteresis model with the deep learning parameter identification algorithm. *J. Magn. Magne. Mater*, **2022**, *562*, 169839, ISSN 0304-8853.
24. Szabó, Z.; Füzi, J. Implementation and identification of Preisach type hysteresis models with Function in closed form. *J. Magn. Magn. Mater.*, **2016**, *406*, 251–258, ISSN 0304-8853.
25. Quondam Antonio, S.; Riganti Fulginei, F.; Laudani, A.; Faba, A.; Cardelli, E. An effective neural network approach to reproduce magnetic hysteresis in electrical steel under arbitrary excitation waveforms. *J. Magn. Magn. Mater.* **2021**, *528*, 167735.
26. Quondam Antonio, S.; Riganti Fulginei, F.; Lozito, G.M.; Faba, A.; Salvini, A.; Bonaiuto, V.; Sargeni, F. Computing Frequency-Dependent Hysteresis Loops and Dynamic Energy Losses in Soft Magnetic Alloys via Artificial Neural Networks. *Mathematics* **2022**, *10*, 2346.
27. Quondam Antonio, S.; Bonaiuto, V.; Sargeni, F.; Salvini, A. Neural Network Modeling of Arbitrary Hysteresis Processes: Application to GO Ferromagnetic Steel. *Magnetochemistry* **2022**, *8*, 18.
28. Faba, A.; Riganti Fulginei, F.; Quondam Antonio, S.; Stornelli, G.; Di Schino, A.; Cardelli, E. Hysteresis Modelling in Additively Manufactured FeSi Magnetic Components for Electrical Machines and Drives.*IEEE Trans. Industr. Electr.*, **2024**, *71*, 2188–2197.
29. Vuokila, N.; Cunning, C.; Zhang, J.; Akel, N.; Khan, A.; Lowther, D.A. The Application of Neural Networks to the Modeling of Magnetic Hysteresis. *IEEE Trans. Magn.* **2024**, *60*, 7300604.
30. Akbarzadeh, V.; Davoudpour, M.; Sadeghian, A. Neural network modeling of magnetic hysteresis. In Proceedings of the 2008 IEEE International Conference on Emerging Technologies and Factory Automation, Hamburg, Germany, 15–18 September 2008; pp. 1267–1270.
31. Ding C.; Bai Y.; Ji Y.; Ma P. Neural Network Modeling of Complex Hysteresis Loops in Ferromagnetic Materials. *IEEJ Trans. Electr. Electronic Eng.* **2024**. https://doi.org/10.1002/tee.24194.
32. Chen, G.; Chen, G.; Lou, Y. Diagonal Recurrent Neural Network-Based Hysteresis Modeling. *IEEE Trans. Neural Networks Learn. Syst.*, **2022**, *33*, 7502–7512.
33. Grech, C.; Buzio, M.; Pentella, M.; Sammut, N. Dynamic Ferromagnetic Hysteresis Modelling Using a Preisach-Recurrent Neural Network Model. *Materials*, **2020**, *13*, 2561.
34. Farrokh, M.; Dizaji, F.S.; Dizaji, M.S. Hysteresis Identification Using Extended Preisach Neural Network. *Neural Process Lett.* **2022**, *54*, 1523–1547.
35. Zhang, H.; Yang, C.; Zhang, Y.; Li, Y.; Chen, Y. Temperature-dependent hysteresis model based on temporal convolutional network. *AIP Adv.* **2024**, *14*, 025321 .
36. Chen, S.; Xu, M.; Liu, S.; Liu, H.; Su, L. CNN–AUPI-Based Force Hysteresis Modeling for Soft Joint Actuator. *Arab. J. Sci. Eng.* **2024**, *49*, 14577–14591.
37. Teoh, T.T.; Rong, Z. Deep Convolutional Generative Adversarial Network. In *Artificial Intelligence with Python. Machine Learning: Foundations, Methodologies, and Applications*; Springer: Singapore, 2022.
38. Liu, Y.; Zhou, R.; Huo, M. Long short term memory network is capable of capturing complex hysteretic dynamics in piezoelectric actuators. *Electron. Lett.* **2019**, *55*, 80–82.
39. Chandra, A.; Daniels, B.; Curti, M.; Tiels, K.; Lomonova, E.A. Magnetic Hysteresis Modeling with Neural Operators. *arXiv* **2024**, arXiv:2407.03261v1.
40. Shan, T.; Li, M. Deep learning techniques for electromagnetic forward modeling. In *Applications of Deep Learning in Electromagnetics: Teaching Maxwell's Equations to Machines*; Li, M., Salucci, M., Eds.; Scitech Publishing: Raleigh, NC, USA, 2024.
41. Cesay, S.;Teng, P.; Wang, R.; Yue, H.; Khan, A.; Lowther, D. Generalizable DNN based multi-material Hysteresis Modelling. In Proceedings of the 2022 IEEE 20th Biennial Conference on Electromagnetic Field Computation (CEFC), Denver, CO, USA, 24–26 October 2022.
42. Lozito, G.M.; Quercio, M.; Sabino, L.; Laudani, A. A Comparative Analysis on Different Deep Neural Network Models for Magnetic Hysteresis with Distorted Excitation Waveforms. In Proceedings of the 2024 International Conference on Electrical Machines (ICEM), Torino, Italy, 1–4 September 2024; pp. 1–6.
43. Francomano, E.; Paliaga, M. Highlighting numerical insights of an efficient SPH method. *Appl. Math. Comput.* **2018**, *339*, 899–915.
44. Antonelli, L.; Francomano, E.; Gregoretti, F. A CUDA-based implementation of an improved SPH method on GPU. *Appl. Math. Comput.* **2021**, *409*, 125482.
45. Buhmann, M.D. *Radial Basis Functions: Theory and Implementations*; Cambridge University: Cambridge, UK, 2003; ISBN 0-521-63338-9.
46. Chen, S.; Cowan, C.F.N.; Grant, P.M. Orthogonal Least Squares Learning Algorithm for Radial Basis Function Networks. *IEEE Trans. Neural Netw.* **1991**, *2*, 302–309.
47. Marquardt, D.W. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *J. Soc. Industr. Appl. Math.* **1963**, *11*, 431–441.
48. Salvini, A.; Riganti-Fulginei, F.; Coltelli, C. A neuro-genetic and time-frequency approach to macromodeling dynamic hysteresis in the harmonic regime. *IEEE Trans. Magn.* **2003**, *39*, 1401–1404.
49. Zirka, S.E.; Moroz, Y.I. Hysteresis modeling based on similarity. *IEEE Trans. Magn.* **1999**, *35*, 2090–2096.
50. Takács, J. The Everett Integral and Its Analytical Approximation,Advanced Magnetic Materials, Dr. Leszek Malkinski (Ed.), 2012, ISBN: 978-953-51-0637-1, InTech, Available from: http://www.intechopen.com/books/advanced-magnetic-materials/the-everett-integral-and-its-analytic- approximation (accessed on 18 September 2024).
51. Biorci, G.; Pescetti, D. Analytical theory of the behaviour of ferromagnetic materials. *Nuovo Cim.* **1958**, *7*, 829–842.

52. Bernard, Y.; Mendes, E.; Bouillault, F. Dynamic hysteresis modeling based on Preisach model. *IEEE Trans. on Mag.* **2002**, *38*, 885–888.
53. Kadar, G.; Kisdi-Koszo, E.; Kiss, L.; Potocky, L.; Zatroch, M.; Della Torre, E. Bilinear product Preisach modeling of magnetic hysteresis curves. *IEEE Trans. on Mag.* **1989**, *25*, 3931–3933.
54. Goodfellow, I.; Bengio, Y.; Courville, A. *Ch.10: Sequence Modeling: Recurrent and Recursive Nets in Deep Learning*; MIT Press: Cambridge, MA, USA, 2016. Available online: www.deeplearningbook.org (accessed on 18 September 2024).
55. Ala, G.; Catrini, P.; Ippolito, M.G.; La Villetta, M.; Licciardi, S.; Musca, R. Deep Learning for Smart Grid and Energy Context. In Proceedings of the Asia Meeting on Environment and Electrical Engineering (EEE-AM), Hanoi, Vietnam, 13–15 November 2023; pp. 1–6.
56. Chung, J.; Gulcehre, C.; Cho, K.; Bengio, Y. Empirical evaluation of gated recurrent neural networks on sequence modeling, NIPS 2014 Workshop on Deep Learning. *arXiv* **2014**, arXiv:1412.3555.
57. Hagan, M.T.; Menhaj, M.B. Training feedforward networks with the Marquardt algorithm. *IEEE Trans. Neural Netw.* **1994**, *5*, 989–993.
58. Chandra, A.; Kapoor, T.; Daniels, B.; Curti, M.; Tiels, K.; Tartakovsky, D.M.; Lomonova, E.A. Neural oscillators for magnetic hysteresis modeling. *arXiv* **2023**, arXiv:2308.12002.
59. Lecun, Y.; Bengio, Y. Convolutional networks for images, speech, and time series. In *The Handbook of Brain Theory and Neural Networks*, 2nd ed.; Arbib, M.A., Ed.; The MIT Press: Cambridge, MA, USA, 1995; pp. 276–278.
60. Douglas J.S. Convolutional Neural Networks. In *Demystifying Deep Learning: An Introduction to the Mathematics of Neural Networks*; IEEE: Piscataway, NJ, USA, 2024; pp.111–131.
61. Badel, A.; Qiu, J.; Nakano, T. A new simple asymmetric hysteresis operator and its application to inverse control of piezoelectric actuators. *IEEE Trans. Ultrason. Ferroelectr. Freq. Control* **2008**, *55*, 1086–1094.
62. Pollok, S.; Olden-Jorgensen, N.; Jorgensen, P.S.; Bjork, R. Magnetic field prediction using generative adversarial networks. *J. Magn. Magn. Mater.* **2023**, *571*, 170556.
63. He, K.; Zhang, X.; Ren, S.; Sun, J. Deep residual learning for image recognition. In Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition (CVPR), Las Vegas, NV, USA, 27–30 June 2016; pp. 770–778.
64. Wei J.D.; Sun C.T. Constructing hysteretic memory in neural networks. *IEEE Trans. Syst. Man Cybern Part B Cybern* **2000**, *30*, 601–609.
65. Joghataie A.; Farrokh M. Dynamic analysis of nonlinear frames by prandtl neural networks. *J. Eng. Mech.* **2008**, *134*, 961–969.
66. Chiang, D.Y. The generalized Masing models for deteriorating hysteresis and cyclic plasticity. *Appl. Math. Model.* **1999**, *23*, 847–863, ISSN 0307-904X.
67. Farrokh M.; Joghataie A. Adaptive modeling of highly nonlinear hysteresis using preisach neural networks. *J. Eng. Mech.* **2013**, *140*, 06014002.
68. Lu, L.; Jin, P.; Pang, G.; Zhang, Z.; Karniadakis, G.E. Learning nonlinear operators via DeepONet based on the universal approximation theorem of operators. *Nat. Mach. Intell.* **2021**, *3*, 218–229.
69. Chen, T.; Chen, H. Universal approximation to nonlinear operators by neural networks with arbitrary activation functions and its application to dynamical systems. *IEEE Trans. Neural Netw.* **1995**, *6*, 911–917.
70. Azizzadenesheli, K.; Kovachki, N.; Li, Z.; Liu-Schiaffini, M.; Kossaifi, J.; Anandkumar, A. Neural operators for accelerating scientific simulations and design. *Nat. Rev. Phys.* **2024**, *6*, 320–328.