materials

MDPI

Article

Dynamic Ferromagnetic Hysteresis Modelling Using a Preisach-Recurrent Neural Network Model

Christian Grech [1,2,*], Marco Buzio [2], Mariano Pentella [2,3] and Nicholas Sammut [1,2]

[1] Faculty of Information and Communications Technology, University of Malta, MSD2080 Msida, Malta; nicholas.sammut@um.edu.mt

[2] CERN, European Organization for Nuclear Research, 1211 Geneva, Switzerland; Marco.Buzio@cern.ch (M.B.); mariano.pentella@cern.ch (M.P.)

[3] Department of Applied Science and Technology, Politecnico di Torino, 10129 Turin, Italy

* Correspondence: christian.grech.12@um.edu.mt

Received: 10 May 2020; Accepted: 29 May 2020 ; Published: 4 June 2020

Abstract: In this work, a Preisach-recurrent neural network model is proposed to predict the dynamic hysteresis in ARMCO pure iron, an important soft magnetic material in particle accelerator magnets. A recurrent neural network coupled with Preisach play operators is proposed, along with a novel validation method for the identification of the model's parameters. The proposed model is found to predict the magnetic flux density of ARMCO pure iron with a Normalised Root Mean Square Error (NRMSE) better than 0.7%, when trained with just six different hysteresis loops. The model is evaluated using ramp-rates not used in the training procedure, which shows the ability of the model to predict data which has not been measured. The results demonstrate that the Preisach model based on a recurrent neural network can accurately describe ferromagnetic dynamic hysteresis when trained with a limited amount of data, showing the model's potential in the field of materials science.

Keywords: ARMCO pure iron; dynamic hysteresis loop; machine learning; magnetic properties; particle accelerators; Preisach; recurrent neural networks

1. Introduction

The phenomenon of hysteresis occurs when a system is not only dependent on present input values, but also on past input values. The resulting hysteresis loops make modelling a challenge due to the non-linearity exhibited. In phenomena such as ferromagnetism, dynamic effects add a further aspect of complexity. The interaction between electric and magnetic fields results in induced eddy currents, which modify the magnetic hysteresis characteristics. Proposed models to predict hysteresis can be classified into two main categories: physical models and phenomenological models. The physical models are built on the description of the object being modelled using physics laws. Unfortunately, in most cases such models are mathematically complex and require detailed knowledge of the physical properties of the material being modelled. Phenomenological models make use of conventional identification methods, which typically have no physical meaning. Such models are mostly empirical, based on previously acquired experimental data. The most popular phenomenological hysteresis model is the Preisach model [1], used in a vast range of applications. Among the different hysteresis models available in literature, the Preisach-type models prove to have a great potential to explain various magnetization processes, and is one of the most popular models to capture the hysteresis behaviour in non-linear systems.

The biggest challenge in implementing a Preisach model is the identification of the model parameters, because the use of empirical methods makes the procedure problem-dependent [2]. There are numerous identification methods used in literature [3–5]. In most cases, such a function can be identified using measured data and applying a numerical approach, which can be a specific formulation [6,7] or using conventional methods such as an Everett integral [8] or Gaussian/Lorentzian functions [9]. The function can then be applied in software using a look-up table of values. Other 'black-box' identification techniques used alongside the Preisach model include genetic algorithms [10], fuzzy models [11] and artificial neural networks (ANN) [12–14]. In [15], Saliah et al. showed that using an ANN can match results produced by the conventional methods, considerably reducing the time overhead. In [16], Serpico and Visone build an ANN hysteresis model, which is able to model rate-independent hysteresis when combined with Preisach operators as inputs. Similarly, different neural network configurations have been used for modelling the rate-independent hysteresis of magnetic shape memory alloys [17,18].

The Preisach model implementation described above is rate-independent, meaning that the hysteresis output is determined solely by the input's extreme values, and the input ramp rates do not impact the hysteresis loop. Mayergoyz [19] introduced the dependence of the weight function on the speed of output variations; similarly, Mrad and Hu [20] and Song and Li [21] proposed an input-rate dependence of the weight function. Both methods suppose the weight function is the right place to add in dynamic behaviors. On the other hand, in [22] a linear dynamic model is added before the classical Preisach operator and the dynamics are assumed to only happen inside the linear dynamic part. The latter cascade structure can be referred to as an 'external dynamic hysteresis model' and is found in several works [23–25]. In this work, we will consider the rate-dependency without including an additional element. This can be accounted for by implementing the Preisach model's weighting function using dynamic neural networks such as a recurrent neural network (RNN), where each layer has a recurrent connection, allowing the network to have an infinite dynamic response to time series input data. An example of the implementation of such model configurations for hysteretic data can be found in [26], where an Elman RNN was successful in predicting the major loop at several frequencies. In [27], with the addition of Preisach operators at the input stage of an internal time-delay neural network, the major and minor loop hysteresis of a Giant Magnetostrictive Actuator was modelled successfully, however this work did not demonstrate functionality in the saturation regions.

In this paper, a Preisach model is implemented using a single RNN, which is able to predict the different dynamic hysteresis loops of ferromagnetic materials, when a limited amount of measurement data is available. In particular, the model is trained using three particular frequencies and tested on a data set which consists of a different frequency. As a novel contribution to the current research state, both major and minor loop hysteresis are investigated, including saturation regions. The motivation behind this work lies in the prediction of the dynamic behaviour of materials used in the manufacturing of magnets for particle accelerators. As an example, ARMCO is used as a test material being used as part of the superconducting magnets for the High Luminosity upgrade of the Large Hadron Collider at the European Nuclear Research Centre (CERN) [28,29], where pulsed fields are employed. Hence, the knowledge of the material's dynamic behaviour, which is not easily modelled, is required and this work attempts to propose an accurate model, trained with a limited amount of measurement data, which can also be used at a moderate computational cost. Whilst a vector hysteresis model is required for a 3D model of an accelerator magnet, the model can be used to describe the vertical field used for the RF control of synchrotrons. The description of the eddy-currents transients when ramping up the material during a test is complicated due to its hysteretic behaviour and the geometry, generally toroidal, which complicates the formulation [30]. This combined with the intrinsic nature of the flux-metric method adopted for the material measurement, introduces an uncertainty component on the results, especially on the coercive field determination [31,32]. The knowledge of the dynamic behaviour of the material can be potentially used to extrapolate the data in DC, allowing the separation of the rate-independent hysteresis from the rate-dependent part, and reducing the overall uncertainty.

In Section 2, the experimental details behind the measurements, the theory behind the model and the validation technique are described. Section 3 describes the results obtained, including the results of a univariate sensitivity analysis. As test material, ARMCO was considered, being an important yoke material in particle accelerators [33]. Moreover, having an electrical conductivitiy in the order of 10^8 S/m and a relative permeability in ther order of 3000, the effect of the eddy currents on the hysteresis loop can be measured with a good signal-to-noise ratio.

2. Materials and Methods

2.1. Experimental Details

The measurements in this work are performed using a split-coil permeameter [28], shown in Figure 1, and performed on toroidal test specimens. The equipment consists of three 90-turn coils, separable by means of an opening mechanism. A slot allows the insertion of the sample to be tested in the permeameter, therefore avoiding the time-demanding operation of a custom coil winding onto the sample [34]. The two outermost coils are used as excitation coils and powered in series up to a maximum current of 40 A, corresponding to a maximum magnetic field of 24 kAm^{-1}. The innermost coil is used to detect the induced voltage. The entire system is cooled by compressed air.

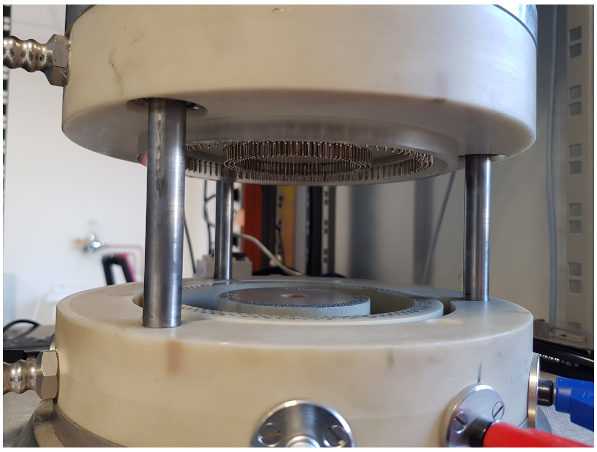

Figure 1. The split-coil permeameter.

The measurements are performed by means of the fluxmetric method, as shown in Figure 2. The test specimen is magnetized by the two excitation coils, having in total N_e = 180 turns and powered in series by a voltage-controlled current generator. Given r_1 and r_2 respectively the inner and the outer radius of the test specimen, the magnetic field is equal to:

$$H(t) = \frac{N_e i(t)}{2\pi r_0} \tag{1}$$

where:

$$r_0 = \frac{r_2 - r_1}{ln\frac{r_2}{r_1}} \tag{2}$$

The magnetic flux density is evaluated by:

$$B(t) = \frac{1}{A_s}\left(\frac{\Phi(t)}{N_s} - \mu_0 H(A_t - A_s)\right) \tag{3}$$

where A_s is the cross-sectional area of the sample, A_t the cross-sectional area of the sensing coil, N_s is the number of turns of the sensing coil, μ_0 the permeability of the free space and $\Phi(t)$ the magnetic flux, evaluated by integrating the induced voltage.

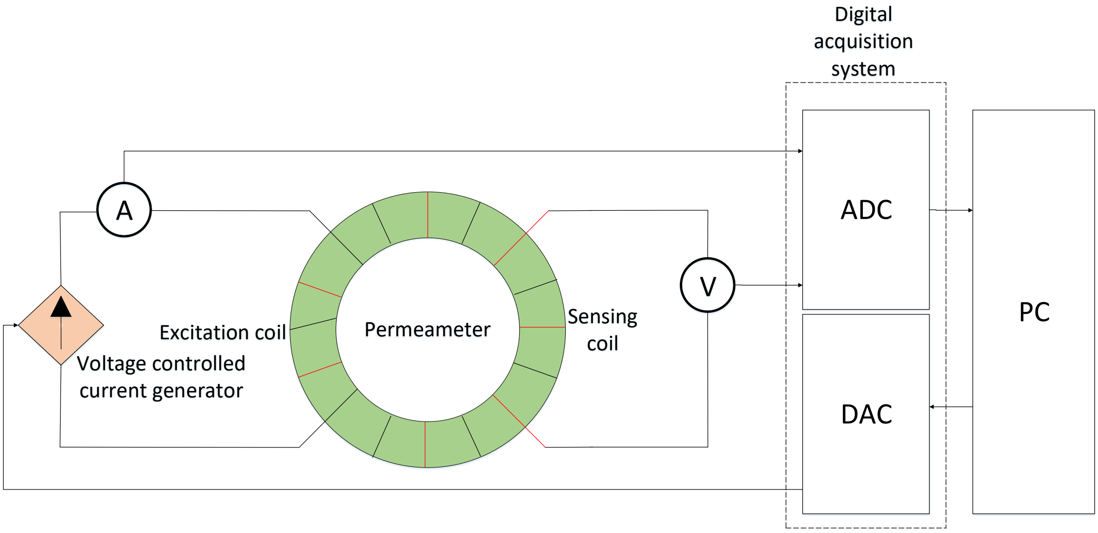
Digital acquisition system
ADC
DAC
PC
A
V
Permeameter
Excitation coil
Sensing coil
Voltage controlled current generator

Figure 2. Measurement system layout.

Both the current and the voltage are acquired by a digital acquisition system (DAQ), a NI 4461 [35] by National Instruments at frequency of 20 kHz. In particular, the value of the current is acquired using a MACC^2 PLUS direct current transducer [36]. The value of the magnetic flux density is acquired with an uncertainty of 0.5 mT whereas the magnetic field is known with an accuracy of 0.1 Am^{-1}. The current is ramped back and forth, between positive and negative symmetric values, and at the end of each ramp the current is kept constant for 0.5 s.

For the major hysteresis loops, the plateau amplitudes are chosen in such a way that the material is brought in saturation state. In order to acquire only the hysteresis loop without including the initial magnetization branch, a pre-cycle is applied before the specific cycle. In the case of the minor loops, the same procedure is carried out, with the difference that the plateau amplitudes at each cycle are increased. Between each acquisition and the following one, the sample is demagnetized. The magnetic flux density, $B(t)$ in different dynamic hysteretic conditions obtained using the magnetic field, $H(t)$ are the source of information for this work. In the case of minor loops, this work is limited to symmetrical, first-order curves.

2.2. Hysteresis Modelling Based on Preisach Memory

In general, the Preisach model is expressed using a double integrator in continuous form as [1]:

$$\hat{y}(t) = \iint\limits_{\alpha \geq \beta} \mu(\alpha, \beta) \gamma_{\alpha\beta} u(t) \mathrm{d}\alpha \mathrm{d}\beta \tag{4}$$

where $\hat{y}(t)$ is the model output at time t, $u(t)$ is the model input at time t, while $\gamma_{\alpha\beta}$ are elementary rectangular hysteresis operators with α and β being the up and down switching values, respectively. These operators can only assume a value of $+1$ or -1. The density function $\mu(\alpha, \beta)$ is a weighting function, which represents the only model unknown which has to be determined from experimental data. In [37,38], following a change in co-ordinates $r = (\alpha - \beta)/2$, $v = (\alpha + \beta)$, $\hat{\mu} = \mu(v + r, v - r)$, it is shown that the boundary between the $+1$ and -1 regions in the Preisach half-plane with coordinates $r > 0$, $v \in \mathbb{R}$, is described by the function $v = P[u](t)$, known as the play operator. This makes it possible to rearrange Equation (4) as:

$$\hat{y}(t) = \int_0^{+\infty} g(r, P[u(t)])\, \mathrm{d}r \tag{5}$$

which can be discretized to n play operators as follows:

$$\hat{y}(t) = \sum_{j=1}^{n} \phi_j P_j[u](t) \tag{6}$$

where ϕ_j represents the density function of the jth play operator, which has to be identified. The play operator is shown in Figure 3, and defined in Equation (7):

$$P_j[u](t) = \max(u(t) - r_j, \min(u(t) + r_j, P_j[u](t-1))) \tag{7}$$

$$P_j[0] = \max(u(0) - r_j, \min(u(0) + r_j, k_0)) \tag{8}$$

where k_0 is the initial condition of the operator and r_j represents the memory depth as follows:

$$r_j = \frac{j-1}{n}[\max(u(t)) - \min(u(t))] \tag{9}$$

where $j = 1, 2, 3, \ldots n$.

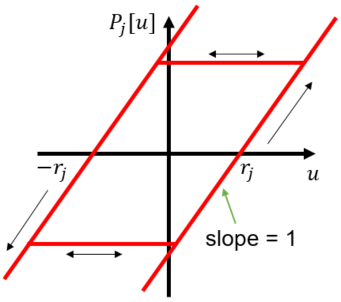

Figure 3. Play operator.

2.3. Identifying ϕ Using Recurrent Neural Networks

Artificial neural networks are able to map non-linear data in various applications. In this case, a recurrent neural network (RNN) will be used to replace the density function ϕ_j of the discrete Preisach model (Equation (6)), as these structures are recognized for their ability to model any non-linear dynamic system, up to a given degree of accuracy [39]. RNNs are distinguished from feed-forward networks by the feedback loop connected to their past decisions, ingesting their own outputs moment after moment as input. This means that such networks can be used to model dynamic characteristics. Sequential information is preserved in the recurrent network's context layer, which manages to span many time steps as it cascades forward to affect the processing of each new example.

In general, a RNN can be seen as a group of nodes, consisting of three different kinds of nodes, namely the input, hidden and output nodes, organized in separate layers, as shown in Figure 4. While the input and output layers consist of feed-forward connections, the hidden layer has recurrent ones. At each time step, t, the input vector $\mathbf{v}(t)$ is processed at the input layer. Each instant $\mathbf{v}(t)$ is summed with the bias vector ${}^{1}\mathbf{b}$ and multiplied by the input weight matrix ${}^{1}\mathbf{w}$. Analogously, the internal state $\mathbf{z}(t)$, delayed by a number of time instants d, is multiplied by the gain factor ${}^{\mathrm{h}}\mathbf{w}$ and added to the input state as follows:

$$\mathbf{z}(t) = f_h[{}^{1}\mathbf{w}(\mathbf{v}(t) + {}^{1}\mathbf{b}) + {}^{\mathrm{h}}\mathbf{w}(\mathbf{z}(t-d))] \tag{10}$$

where $f_h(x)$ is the activation function, in this case a hyperbolic tangent function, which is given as:

$$f_h(x) = \frac{2}{1+e^{-2x}} - 1 \tag{11}$$

The internal state $\mathbf{z}(t)$ is then added with bias ${}^2\mathbf{b}$, multiplied by the weight ${}^2\mathbf{w}$, and the result is passed through a linear activation function $f_o(x)$ as follows:

$$\hat{y}(t) = f_o[{}^2\mathbf{w}(\mathbf{z}(t) + {}^2\mathbf{b})] \tag{12}$$

where $\hat{y}(t)$ is the predicted output at time t.

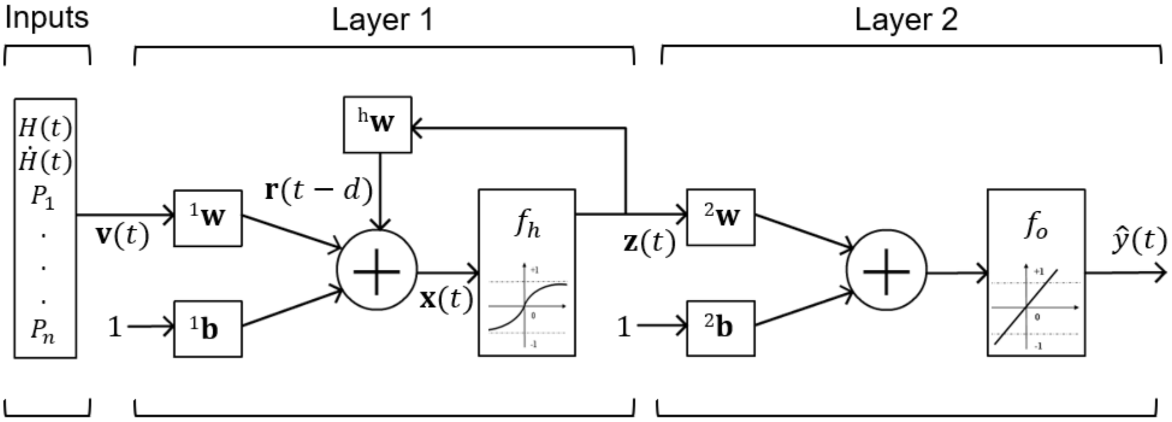

Figure 4. Schematic diagram of the Preisach-RNN model.

The Deep Learning Toolbox [40] by MATLAB is used to determine the weights of the network using the *layrecnet* command. The training algorithm is the Levenberg-Marquadt algorithm [41], which is a non-linear least squares optimization algorithm incorporated into the backpropagation algorithm for training neural networks as demonstrated in detail in [42]. The algorithm aims to optimize the weights according to the following objective function:

$$V(t) = \frac{1}{2}(y(t) - \hat{y}(t))^T(y(t) - \hat{y}(t)) \tag{13}$$

which leads to the update of the weights by the following formula:

$${}^i\mathbf{w}(t+1) = {}^i\mathbf{w}(t) + \eta\left(-\frac{\partial V}{\partial t}\right) \tag{14}$$

where η is a positive number representing the learning rate of the weights.

2.4. Data Analysis Approach

The input data to the model include the magnetic field $H(t)$, its derivative $\dot{H}(t)$ and a number of play operators $P_j[H](t)$ (chosen according to the complexity of the data). All data is normalised to the range of the minor loop data, such that data is approximately in the range $[-1, 1]$. The play operator outputs, P_j, are subsequently calculated based on the normalised magnetic field signal. In this work, the predicted variable is the magnetic flux density measured in Section 2.1, and all measurements are downsampled to 200 Hz.

The data set used for training, validation and testing comprises of three minor loops and three major loops ramping at 1025, 1554 and 6135 Am^{-1}s^{-1}, as shown in Figure 5. This data is divided into three subsets in an interleaved manner. The first subset is the training set (70% of the data), which is used for updating the network weights and biases. The second subset is the validation set (15% of the data), used to decide when to stop training the model, whilst the final set is the testing set (15% of the data), which is used to select the best model structure. Once the network is trained, and the best model structure is chosen, an evaluation signal is used to demonstrate the performance of the model.

The evaluation data set consists of three major loops ramping at 3067 $\mathrm{Am^{-1}s^{-1}}$ and one minor loop ramping at various random ramp-rates between 1000 and 6200 $\mathrm{Am^{-1}s^{-1}}$, as shown in Figure 6.

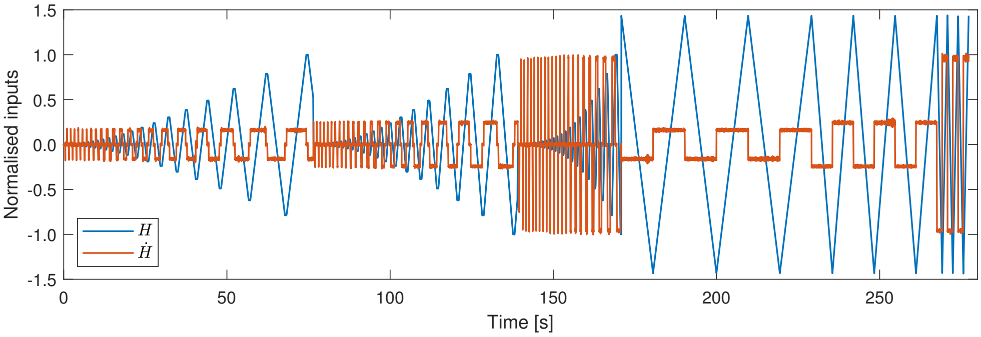

Figure 5. Magnetic field data and its derivative used for training, validation and testing the model.

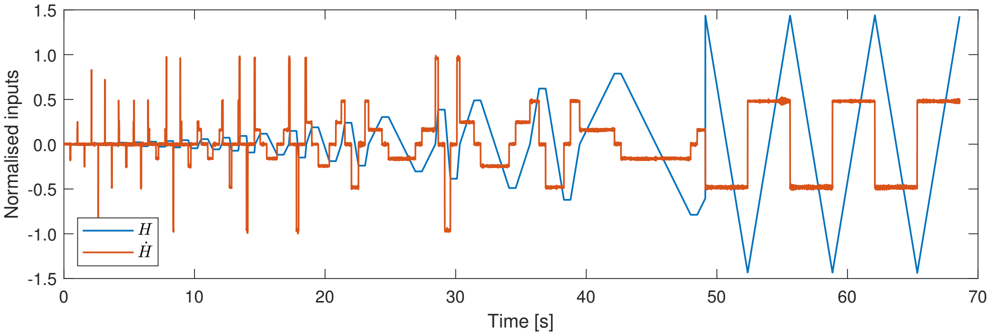

Figure 6. Magnetic field data and its derivative used for evaluating the model.

2.5. Model Validation

Model validation is the process of choosing the best model parameters and making sure that the model is robust to new data. The optimal number of nodes in the hidden layer, h is chosen by doing a search over a range of values. The complete validation process explained below consists of three loops, and is represented by a flowchart in Figure 7.

In the innermost loop, each neural network is trained using the training set as described in Section 2.3 over a certain number of repetitions called epochs. The number of epochs is determined using a method called early stopping [43,44]. In this technique, the error on the validation set is monitored during the training process. The validation error normally decreases during the initial phase of training, as does the training set error. However, when the network begins to overfit the data, the error on the validation set typically begins to rise. When the validation error increases for a specified number of iterations, the training is stopped, and the weights and biases at the minimum of the validation error are returned. Finally, the testing set is used to obtain the performance of the particular network.

Each time a neural network is trained, a different solution is obtained due to different and random initial weight and bias values. As a result, different neural networks trained on the same problem can give different outputs for the same input. To ensure that a neural network of good accuracy has been found, each network is retrained for a number of times, N, which in this work is 10.

In the outermost loop, the number of hidden nodes is varied over the range starting from h_{min} to h_{max} with increments of dh. The network with the best test performance having a number of hidden nodes h, is hence chosen and used with the evaluation data set. The complete list of parameters used in this work is given in Table 1.

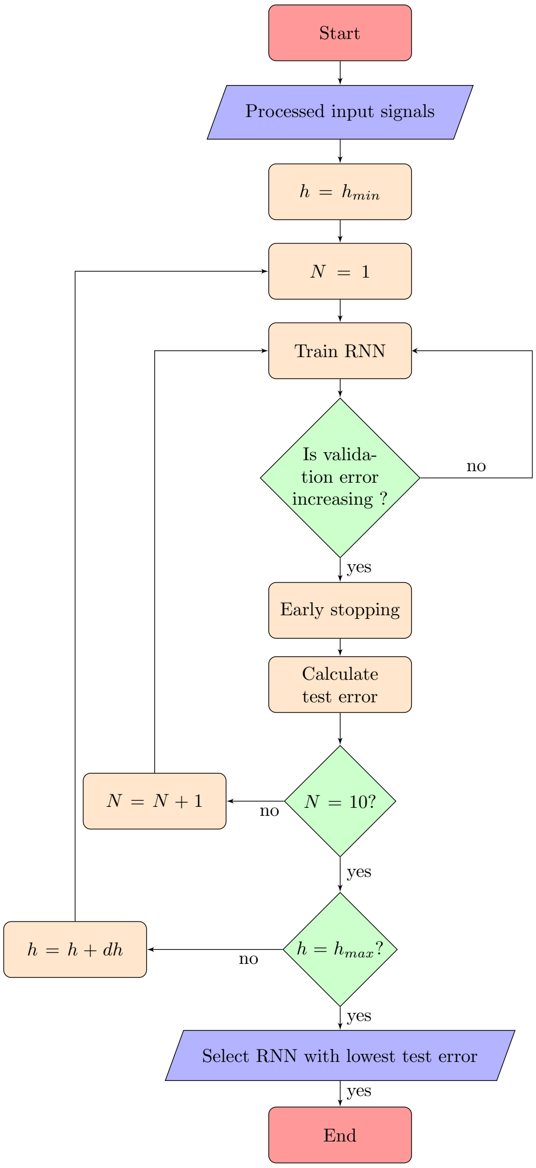

Figure 7. Flowchart showing the model training and validation procedure.

Table 1. RNN parameters list.

	Parameter	**Value**
Model properties	Activation function (hidden layer)	sigmoid
	Activation function (output layer)	linear
	delay, d	2
	hidden nodes search range, $h_{min} : dh : h_{max}$	4:1:13
	number of hidden nodes, h	12
	training repetitions, N	10
Input	No. of play operators	6
Training set	data	70% of data set
	epochs	$\leq$1000
	algorithm	Levenberg-Marquadt
Validation set	data	15% of data set
Testing set	data	15% of data set
Evaluation set	data	unseen data
	metric	NRMSE

2.6. Performance Indicator

The performance indicator used in this work to represent the error between the model and experimental measurements is the normalised root mean square error (NRMSE):

$$\mathrm{NRMSE}(y,\hat{y}) = \frac{1}{\max(y)-\min(y)}\sqrt{\frac{1}{N}\sum_{i=1}^{N}(y_i-\hat{y})_i^2} \tag{15}$$

where y is the actual value, $\hat{y}$ is the modelled quantity and N is the number of samples considered. By normalising the root mean square error, the errors can be scaled relatively to the range of the measurement, thus allowing an appropriate comparison across different conditions.

3. Results

Results from the model training, testing and sensitivity analysis are presented and discussed in this section.

3.1. Minor and Major Loop Model Prediction

Following the model validation and training procedure explained in the previous section, the best performing model is used for predicting the data in the evaluation set shown in Figure 6. In order to get rid of the initial transient phase of the model, the first few predicted samples in the set are discarded. In the case of the major loop, a NRMSE of 0.58% is noted in predicting a hysteresis loop with a new ramp-rate not used for training. Figure 8 shows the major loop data used for training and evaluating the model, as well as the predicted data. For the minor loop data, a NRMSE of 0.66% is noted with different random ramp-rates, including ramp-rates not used for training the model. The magnetic flux density as a function of time is shown in Figure 9.

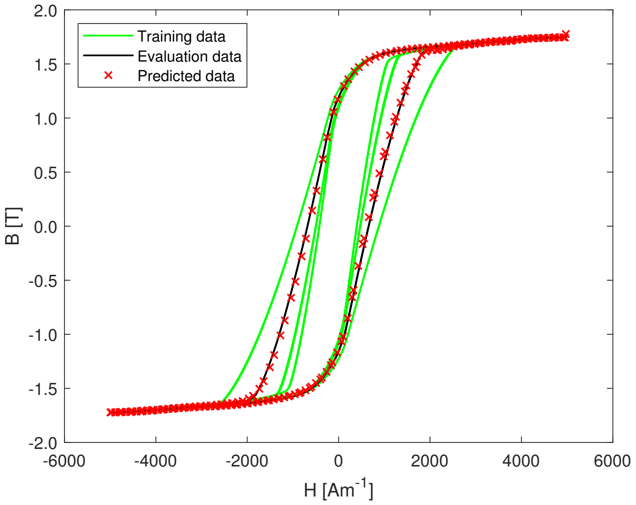

Figure 8. Predicted major loop and experimental data used in training the model.

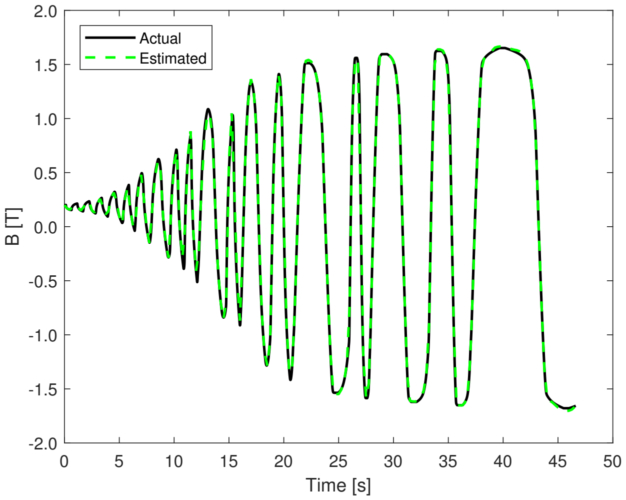

Figure 9. Predicted minor loop with random ramp-rates.

3.2. Effect of Preisach Operators on Performance

In order to quantify the impact of Preisach operators as part of the model, the model's training and validation procedure is repeated without the P_j inputs. Comparing the performance of the two models, the inclusion of the Preisach operators is noted to improve performance by 19% in the case of major loop hysteresis, and 44% in the case of minor loop hysteresis. It has to be noted however, that a model without Preisach operators is computationally less expensive, as the optimal structure contains 241 weights versus 409 weights in the original model. Hence, in the end, a compromise must be made between accuracy and speed according to the model's application.

3.3. Univariate Sensitivity Analysis

A univariate sensitivity analysis provides information on how robust a model is when the input values are varied over a specific range [45]. This analysis is performed to understand which model input variables impact the prediction most significantly, especially since the magnetic field derivative signal, $\dot{H}(t)$ is noisy. A neural network which picks up minor noise during training can overfit the noise as if it is the signal, leading to poor accuracy during validation [46]. One way to confirm this is by performing a sensitivity analysis, where one input parameter is fed a changing signal whilst keeping the other inputs constant, and checking the deviation in the output signal. This is repeated for all the model inputs.

In this exercise, once the model is trained, an input vector is defined and set as 0. Then, for one of the eight variables, a value is assigned between the $[-1,1]$ range. This is repeated for 10,000 samples for each input variable. The predicted output, $\hat{y}_s$ is saved and the standard deviation $\sigma_i(\hat{y}_s)$ for each variable i is calculated. The bar chart in Figure 10 illustrates these results. The analysis results show a higher variation in the predicted output for play operator input parameters, thus the model is more sensitive to changes in these particular variables. These results also confirm the fact that even though the model is trained with a noisy $\dot{H}$ signal, the predicted output is not particularly sensitive to perturbations from this parameter.

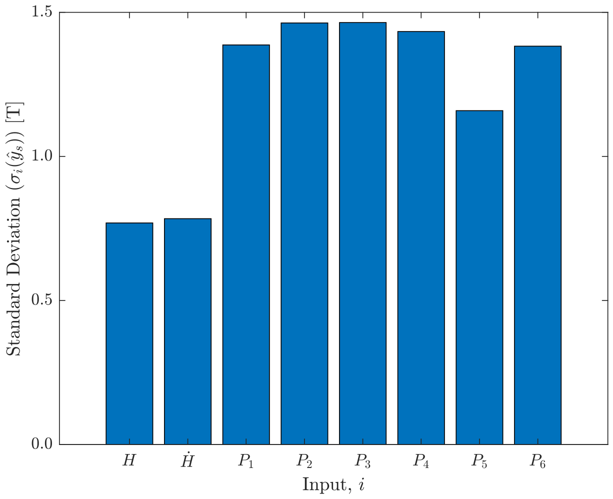

Figure 10. Sensitivity analysis results for each individual input.

4. Conclusions

A Preisach-RNN model is proposed to predict the dynamic characteristics of ferromagnetic materials that does not require *a priori* knowledge of the material and its microstructural behaviour. The model is based on the Preisach memory block where the density function is represented by a recurrent neural network. A thorough training and validation procedure is proposed for the neural network, in order to optimize the weight parameters. We have demonstrated, using ARMCO pure iron measurements, that such a model can predict both major and minor loop dynamic ferromagnetic hysteresis behaviour allowing researchers to estimate the dynamic effects of the material knowing only six different examples at three frequencies. Comparing the model's predictions to experimental data, the model's NRMSE is noted to be better than 0.7%. Moreover, these results prove that the model generalises well to new data, and can potentially be used at different frequencies than the frequencies used in this work. In validating the performance of the model, the positive impact of Preisach operators is noted, even though this comes at a computational cost, which has to be evaluated for each specific industry. Results from a univariate sensitivity analysis also demonstrate that the play operator inputs impact the predicted magnetic flux output most significantly. We believe that this model, predicting major and minor loop hysteresis under different dynamic conditions can be applied to other dynamic hysteresis prediction problems in the realm of materials science.

Author Contributions: Conceptualization, C.G., M.B., M.P. and N.S.; methodology, C.G. and M.P.; software, C.G., and M.P.; validation, C.G.; formal analysis, C.G. and M.P.; investigation, C.G. and M.P.; data curation, C.G. and M.P.; writing—original draft preparation, C.G. and M.P.; writing—review and editing, C.G., M.B., M.P. and N.S.; visualization, C.G.; supervision, M.B. and N.S. All authors have read and agreed to the published version of the manuscript.

Funding: The publication fee was funded by CERN.

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

Abbreviations

The following abbreviations are used in this manuscript:

ANN	Artificial Neural Network
CERN	European Organization for Nuclear Research
NRMSE	Normalised Root Mean Square Error
RNN	Recurrent Neural Network

References

1. Mayergoyz, I. (Ed.) Chapter 1—The Classical Preisach Model of Hysteresis. In *Mathematical Models of Hysteresis and Their Applications*; Electromagnetism; Elsevier Science: New York, NY, USA, 2003; pp. 1–63, doi:10.1016/B978-012480873-7/50002-5. [CrossRef]
2. Saliah, H.; Lowther, D. The use of neural networks in magnetic hysteresis identification. *Phys. B Condens. Matter* **1997**, *233*, 318–323. doi:10.1016/S0921-4526(97)00316-5. [CrossRef]
3. Iyer, R.V.; Tan, X. Control of hysteretic systems through inverse compensation. *IEEE Control Syst. Mag.* **2009**, *29*, 83–99, doi:10.1109/MCS.2008.930924. [CrossRef]
4. Sutor, A.; Rupitsch, S.J.; Lerch, R. A Preisach-based hysteresis model for magnetic and ferroelectric hysteresis. *Appl. Phys. A* **2010**, *100*, 425–430, doi:10.1007/s00339-010-5884-9. [CrossRef]
5. Stakvik, J. Identification, Inversion and Implementaion of the Preisach Hysteresis Model in Nanopositioning. Master's Thesis, Nowegian University of Science and Technology, Trondheim, Norway, 2014.
6. Biorci, G.; Pescetti, D. Analytical theory of the behaviour of ferromagnetic materials. *Il Nuovo Cimento (1955–1965)* **1958**, *7*, 829–842, doi:10.1007/BF02745588. [CrossRef]
7. Ruderman, M.; Bertram, T. Identification of Soft Magnetic B-H Characteristics Using Discrete Dynamic Preisach Model and Single Measured Hysteresis Loop. *IEEE Trans. Magn.* **2012**, *48*, 1281–1284, doi:10.1109/TMAG.2011.2172931. [CrossRef]
8. Kozek, M.; Gross, B. Identification and Inversion of Magnetic Hysteresis for Sinusoidal Magnetization. *Int. J. Online Biomed. Eng.* **2005**. Available online: https://online-journals.org/index.php/i-joe/article/view/299/2990 (accessed on 3 June 2020).
9. Rouve, L.; Waeckerle, T.; Kedous-Lebouc, A. Application of Preisach model to grain oriented steels: Comparison of different characterizations for the Preisach function p(/spl alpha/, /spl beta/). *IEEE Trans. Magn.* **1995**, *31*, 3557–3559, doi:10.1109/20.489568. [CrossRef]
10. Hergli, K.; Marouani, H.; Zidi, M.; Fouad, Y.; Elshazly, M. Identification of Preisach hysteresis model parameters using genetic algorithms. *J. King Saud Univ.-Sci.* **2017**, doi:10.1016/j.jksus.2017.11.005. [CrossRef]
11. Natale, C.; Velardi, F.; Visone, C. Identification and compensation of Preisach hysteresis models for magnetostrictive actuators. *Phys. B Condensed Matter* **2001**, *306*, 161–165, doi:10.1016/S0921-4526(01)00997-8. [CrossRef]
12. Adly, A.; El-Hafiz, S.A. Using neural networks in the identification of Preisach-type hysteresis models. *IEEE Trans. Magn.* **1998**, *34*, 629–635, doi:10.1109/20.668057. [CrossRef]
13. 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.
14. Firouzi, M.; Shouraki, S.B.; Zakerzadeh, M.R. Hysteresis nonlinearity identification by using RBF neural network approach. In Proceedings of the 2010 18th Iranian Conference on Electrical Engineering, Isfahan, Iran, 11–13 May 2010, doi:10.1109/iraniancee.2010.5506985. [CrossRef]
15. Saliah, H.H.; Lowther, D.A.; Forghani, B. A neural network model of magnetic hysteresis for computational magnetics. *IEEE Trans. Magn.* **1997**, *33*, 4146–4148, doi:10.1109/20.619691. [CrossRef]
16. Serpico, C.; Visone, C. Magnetic hysteresis modeling via feed-forward neural networks. *IEEE Trans. Magn.* **1998**, *34*, 623–628, doi:10.1109/20.668055. [CrossRef]
17. Xu, R.; Zhou, M. Elman Neural Network-Based Identification of Krasnosel'skii–Pokrovskii Model for Magnetic Shape Memory Alloys Actuator. *IEEE Trans. Magn.* **2017**, *53*, 1–4, doi:10.1109/TMAG.2017.2700479. [CrossRef]
18. Zhou, M.; Zhang, Q. Hysteresis Model of Magnetically Controlled Shape Memory Alloy Based on a PID Neural Network. *IEEE Trans. Magn.* **2015**, *51*, 1–4, doi:10.1109/tmag.2015.2434933. [CrossRef]
19. Mayergoyz, I.D. Dynamic Preisach models of hysteresis. *IEEE Trans. Magn.* **1988**, *24*, 2925–2927, doi:10.1109/20.92290. [CrossRef]
20. Mrad, R.B.; Hu, H. Dynamic modeling of hysteresis in piezoceramics. In Proceedings of the 2001 IEEE/ASME International Conference on Advanced Intelligent Mechatronics, Proceedings (Cat. No.01TH8556), Como, Italy, 8–12 July 2001, doi:10.1109/aim.2001.936515. [CrossRef]
21. Song, D.; Li, C.J. Modeling of piezo actuator's nonlinear and frequency dependent dynamics. *Mechatronics* **1999**, *9*, 391–410, doi:10.1016/S0957-4158(99)00005-7. [CrossRef]

22. Füzi, J. Computationally efficient rate dependent hysteresis model. *COMPEL-Int. J. Comput. Math. Electr. Electron. Eng.* **1999**, *18*, 445–457, doi:10.1108/03321649910275080. [CrossRef]
23. Kuczmann, M. Dynamic Preisach model identification applying FEM and measured BH curve. *COMPEL-Int. J. Comput. Math. Electr. Electron. Eng.* **2014**, *33*, 2043–2052, doi:10.1108/COMPEL-11-2013-0368. [CrossRef]
24. Makaveev, D.; Dupré, L.; Wulf, M.D.; Melkebeek, J. Dynamic hysteresis modelling using feed-forward neural networks. *J. Magn. Magn. Mater.* **2003**, *254–255*, 256–258. doi:10.1016/S0304-8853(02)00785-0. [CrossRef]
25. Tan, X.; Baras, J.S. Modeling and control of hysteresis in magnetostrictive actuators. *Automatica* **2004**, *40*, 1469–1480, doi:10.1016/j.automatica.2004.04.006. [CrossRef]
26. Saghafifar, M.; Kundu, A.; Nafalski, A. Dynamic magnetic hysteresis modelling using Elman recurrent neural network. *Int. J. Appl. Electromagn. Mech.* **2001**, *13*, 209–214. [CrossRef]
27. Wang, Q.; Su, C.Y.; Tan, Y. On the Control of Plants with Hysteresis: Overview and a Prandtl-Ishlinskii Hysteresis Based Control Approach. *ACTA Autom. Sin.* **2005**. [CrossRef]
28. Parrella, A.; Arpaia, P.; Buzio, M.; Liccardo, A.; Pentella, M.; Principe, R.; Ramos, P. Magnetic Properties of Pure Iron for the Upgrade of the LHC Superconducting Dipole and Quadrupole Magnets. *IEEE Trans. Magn.* **2018**, *55*, 1–4. [CrossRef]
29. Arpaia, P.; Buzio, M.; Bermudez, S.I.; Liccardo, A.; Parrella, A.; Pentella, M.; Ramos, P.M.; Stubberud, E. A Superconducting Permeameter for Characterizing Soft Magnetic Materials at High Fields. *IEEE Trans. Instrum. Meas.* **2019**. [CrossRef]
30. Namjoshi, K.; Lavers, J.D.; Biringer, P. Eddy-current power loss in toroidal cores with rectangular cross section. *IEEE Trans. Magn.* **1998**, *34*, 636–641. [CrossRef]
31. Anglada, J.; Arpaia, P.; Buzio, M.; Pentella, M.; Petrone, C. Characterization of Magnetic Steels for the FCC-ee Magnet Prototypes.
32. Grossinger, R.; Mehboob, N.; Suess, D.; Turtelli, R.S.; Kriegisch, M. An eddy-current model describing the frequency dependence of the coercivity of polycrystalline Galfenol. *IEEE Trans. Magn.* **2012**, *48*, 3076–3079. [CrossRef]
33. Sgobba, S. Physics and measurements of magnetic materials. In Proceedings of the CERN Accelerator School CAS 2009: Specialised Course on Magnets, Bruges, Belgium, 16–25 June 2009. doi:10.5170/CERN-2010-004.39. [CrossRef]
34. Arpaia, P.; Buzio, M.; Fiscarelli, L.; Montenero, G.; Walckiers, L. High-performance permeability measurements: A case study at CERN. In Proceedings of the 2010 IEEE Instrumentation & Measurement Technology Conference Proceedings, Austin, TX, USA, 3–6 May 2010; pp. 58–61.
35. National Instruments. NI 446x Specifications. 2008. Available online: http://www.ni.com/pdf/manuals/373770j.pdf (accessed on 3 June 2020).
36. Hitec. MACC 2 Plus. 2016. Available online: http://www.pm-sms.com/files/2016/02/Specifications-MACC-2-plus.pdf (accessed on 3 June 2020).
37. Krejčí, P. On Maxwell equations with the Preisach hysteresis operator: The one-dimensional time-periodic case. *Apl. Mat.* **1989**, *34*, 364–374.
38. Brokate, M. Some mathematical properties of the Preisach model for hysteresis. *IEEE Trans. Magn.* **1989**, *25*, 2922–2924, doi:10.1109/20.34325. [CrossRef]
39. Schäfer, A.M.; Zimmermann, H.G. Recurrent Neural Networks are universal approximators. *Int. J. Neural Syst.* **2007**, *17*, 253–263, doi:10.1142/S0129065707001111. [CrossRef]
40. MATLAB. Deep Learning Toolbox. 2019. Available online: https://www.mathworks.com/help/deeplearning/index.html?s_tid=mwa_osa_a (accessed on 3 June 2020).
41. Marquardt, D. An Algorithm for Least-Squares Estimation of Nonlinear Parameters. *J. Soc. Ind. Appl. Math.* **1963**, *11*, 431–441, doi:10.1137/0111030. [CrossRef]
42. Hagan, M.T.; Menhaj, M.B. Training feedforward networks with the Marquardt algorithm. *IEEE Trans. Neural Netw.* **1994**, *5*, 989–993. [CrossRef] [PubMed]
43. Prechelt, L. Early Stopping—But When? In *Neural Networks: Tricks of the Trade*, 2nd ed.; Springer: Berlin/Heidelberg, Germany, 2012; pp. 53–67, doi:10.1007/978-3-642-35289-8_5. [CrossRef]
44. Yao, Y.; Rosasco, L.; Caponnetto, A. On Early Stopping in Gradient Descent Learning. *Constr. Approx.* **2007**, *26*, 289–315, doi:10.1007/s00365-006-0663-2. [CrossRef]

45. Salciccioli, J.D.; Crutain, Y.; Komorowski, M.; Marshall, D.C. Sensitivity Analysis and Model Validation. In *Secondary Analysis of Electronic Health Records*; Springer International Publishing: Cham, Switzerland, 2016; pp. 263–271, doi:10.1007/978-3-319-43742-2_17. [CrossRef]

46. Liu, Z.P.; Castagna, J.P. Avoiding overfitting caused by noise using a uniform training mode. In Proceedings of the IJCNN'99—International Joint Conference on Neural Networks. Proceedings (Cat. No.99CH36339), Washington, DC, USA, 10–16 July 1999; Volume 3, pp. 1788–1793, doi:10.1109/IJCNN.1999.832649. [CrossRef]