Prediction of Volumetric Loss of any Magnetic Material using Deep Neural Network - MagNet

Dixant Bikal Sapkota, Puskar Neupane, Mecon Joshi, Shahabuddin Khan

***Abstract*—This paper outlines the methodology for predicting power loss in magnetic materials. It starts by introducing the concept of core loss and the complexity of modeling it. Steinmetz's Equation is presented to calculate power loss based on frequency and magnetic flux density, but its limitations are highlighted. As an alternative, neural network based method is introduced. The proposed methodology adopts a Long Short-Term Memory Network, expressing the core loss as a function of magnetic flux density (B), frequency (f), temperature (T) and wave classification. Fast Fourier Transform was implemented to reduce the data points of the sampled B waveform while preserving its wave characteristics. The analysis in frequency domain enabled to streamline the training of the dataset. The data set was arranged as required, and the network architecture was designed with appropriate layers and optimal activation functions. Through training on the dataset, the network assimilated the intricate relationships between input variables and known power loss. Evaluation and validation metrics were subsequently employed to gauge the performance of the trained network. This innovative methodology aims to improve the precision of power loss predictions, providing valuable insights into the nuanced behavior of magnetic materials.**

***Index Terms*—MagNet, Frequency Domain, LSTM, FFT**

I. Introduction

MAGNETIC materials are typically characterized by their magnetic domain existence. In the absence of an external magnetizing field (H), these domains are aligned randomly. However, in the presence of such a field, these domains adjust themselves to align in parallel with the direction of the applied field. When an alternating current is applied, the domain's alignment oscillates with the applied field frequency, causing core loss, a phenomenon where heat is generated due to repeated changes in direction and magnitude of the domains.

Charles Steinmetz first developed the equation to characterize the magnetic core loss in 1892. It is an empirical formula that is used to compute the overall losses of power in any magnetic material expressed as power loss per unit volume. It was later modified to include frequency term and named as Steinmetz's Equation [1]. The equation is mathematically expressed as:

$$P_{\mathrm{v}} = k f^{\mathrm{a}} B^{\mathrm{b}} \tag{1}$$

Here, P_{v} is the average power loss per unit volume, f is the frequency, B is the peak value of magnetic flux density and k, a, and b are Steinmetz constants whose values are obtained through B-H curve fitting of the given material. This equation gives a simple understanding of core losses and is a decent equation for general design of equipment. But in the field of power electronics, it is perhaps the weakest link due to its ignorance of the impact of various other factors such as waveform shape, DC bias, ambient and core temperatures, etc. on power losses.

Over the years following, various models were developed by improving upon the Steinmetz equation. For instance, MSE, GSE and iGSE improved the results under non-sinusoidal excitation. Particularly, the improved generalized Steinmetz Equation (iGSE) has been considered an excellent calculation method. However, it gives large error for very high or very low duty cycles [2]. Similarly, other methods such as EEL and WcSE give high errors when duty cycle is far from 0.5 and methods such as DNSE, i^2GSE, RESE and FGSE introduce new parameters which are not typically available in data sheets [2]. This shows the complexity of the process of modeling magnetic core loss.

In recent times, models based on neural networks are being considered for magnetic core loss prediction due to their effectiveness in solving non-linear problems. These models are data based in nature and can predict losses for any given material subjected to a certain field density with known frequency, temperature and additional related features. This approach requires careful study of impacting parameters, design of appropriate model architecture, large and accurate data set and optimal training process. This is the approach taken in this paper. Utilizing the power of the Python programming language and a large number of available libraries such as $numpy$, $pandas$, $tensorflow$, and $keras$, a model has been designed that can predict losses for any magnetic material with decent accuracy and a relatively small number of input features.

II. Preliminary Works/Literature Review

Previous studies in this field have made essential datasets available for the model's training, validation, and testing [3], [4], [5], significantly contributing to the preparatory phases of this study. These datasets assume a pivotal role in refining and evaluating the model.

[6] surveys various publications, exploring the application of AI in Power Electronics to enhance accuracy and efficiency in the sector. In Paper [7], the datasheets for loss and B-H loop utilizing neural networks are shown, along with an illustration of how beneficial neural networks are in forecasting core losses under various operating conditions. The DNN method was implemented in Paper [8], which accurately modeled the core losses and recommended capturing additional scenarios to enhance the model. The most advanced neural network models have also been studied, and they have shown to be more accurate in predicting core losses than traditional machine learning

models. The transformer neural network model described in Paper [9] has proven to be effective in simulating hysteresis loops by symbolically integrating sequence inputs and scalar inputs.

III. Fourier Transform and FFT

Fourier Transform [10] is a mathematical technique which decomposes any signal which is a function of time into its various constituent frequency components. This is a transformation between time-domain representation and frequency-domain representation.

Fast Fourier Transform, abbreviated as FFT, is a mathematical algorithm that computes the Discrete Fourier Transform [11]. The "numpy" library in python provides functionality for calculating one dimensional n-point discrete fourier transform with the FFT algorithm. The input to the FFT algorithm is an array of sequential data points representing a sampled time series data and its result is an array of complex numbers representing the samples of frequency domain spectrum. The Fast Fourier Transform (FFT) has been used here to decrease the computational complexity of the model by reducing the input parameters required for calculation.

IV. Methodology

A. Data Processing

The provided training dataset consists of five different comma separated values (csv) files for every material, each containing data for magnetic field density (B), applied magnetizing field force (H), frequency (F), temperature (T) and measured volumetric loss (V). The number of rows of data is not same for all materials and is listed in Table I. There are 1024 time instance values for B and H in a single cycle and a single value for others. Additional data was provided for the testing of the trained model.

TABLE I
Total number of Provided data for each materials

Material	Training Data	Testing data	Type
3C90	40713	5000	Known
3C94	40068		
3E6	6996		
3F4	6564		
77	11444		
78	11380		
N27	11396		
N30	8978		
N49	8602		
N87	40616		
A	2432	7651	Unknown
B	7400	3172	
C	5357	5357	
D	580	7299	
E	2013	3738	

To reduce the computational time for training the models, there was a need to minimize the data size while preserving as much valuable information as possible to sustain accuracy. To achieve this, the focus was primarily set on the aggregation of the given waveform data. Various methods were explored, including selecting only a few of the numerous provided time instance values, averaging the values over a specific range, and decomposing the wave into its constituent harmonics but the latter seemed to give the most favorable results with higher consistency.

Fast Fourier Transform, shortened as FFT, was employed to transform each of the provided time domain waveform into its frequency domain representation. Specifically, the DC offset and the next 31 harmonics (i.e. first 32 terms in total) were used for model training. This technique preserved the wave characteristics during the training while also reducing the overall data input directly to the model. Since the output of FFT is a complex quantity, its absolute value was considered for each harmonic. This was done because if both real and imaginary values are to be used for model training, there would be twice as much data, which would increase the computational needs. The model's accuracy was observed to be largely unaffected by the inclusion of more harmonics; so, the terms were restricted to 32.

Additionally, further techniques were integrated to efficiently recognize and classify various waveforms. The waves were categorized into sinusoidal, triangular and trapezoidal waves based on their frequency domain representations. Sine waves have high fundamental component and negligible harmonics thereafter, while triangular waves have low fundamental component to second harmonic ratio and trapezoidal waves have high value of the same ratio. These properties were used for the classification and it seemed to give an acceptable classification although there were some outliers in trapezoidal and triangular category. Practically, sinusoidal waves were isolated first, and then waves having fundamental to second harmonic ratio less than 7 were classified as triangular while more than that were classified as trapezoidal. An example of this can be seen in Figure 1.

The goal is to establish a cost-effective equilibrium between model accuracy and resource efficiency through the reduction of input data size and the proper classification of waveforms. After these calculations, the data was re-arranged in a separate array with the labels as shown in Table II.

TABLE II
Data arrangement labels

S.N	B1, B2, B3,... up-to B32	Freq	Temp	Loss	Classification

B. Data Visualization

In order to fully understand the nature of the parameters for each material, various plots were drawn. A scatter plot drawn between different features for 3C90 can be seen in Figure 2. It can be observed that the data doesn't have any defined correlation. High volumetric losses exist for lower B_{max}, but even at higher fields volumetric loss ranges significantly due to changes in frequency and temperature values. From Figure 2(d) it can be observed that high frequencies are only present for lower field densities. This is proved by the correlation between B_{max} and Frequency being -0.430108. The effect of this can be observed in Figure 2(b) where, as we proceed

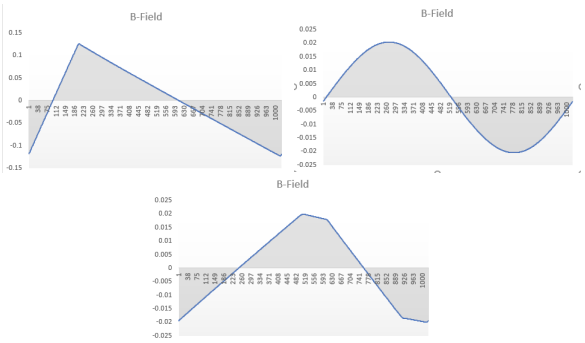

Fig. 1. Classification of magnetic field density waveforms: 0 for triangular, 1 for sinusoidal and 2 for trapezoidal

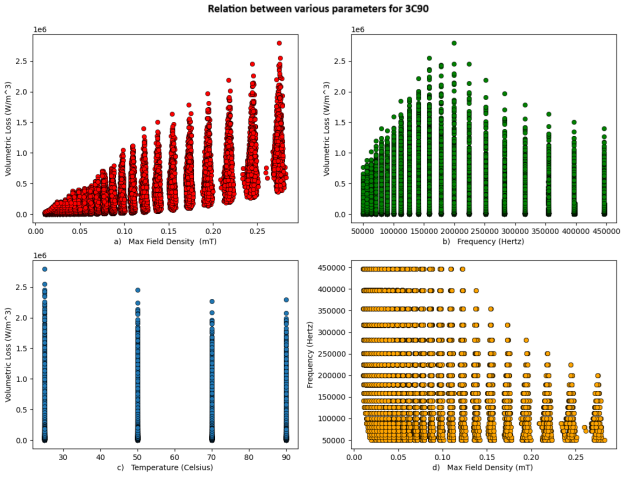

Fig. 2. Scatter plot between various parameters for Material 3C90: a) Maximum Field Density vs Volumetric Loss, b) Frequency vs Volumetric Loss, c) Temperature vs Volumetric Loss, d) Maximum Field Density vs Frequency

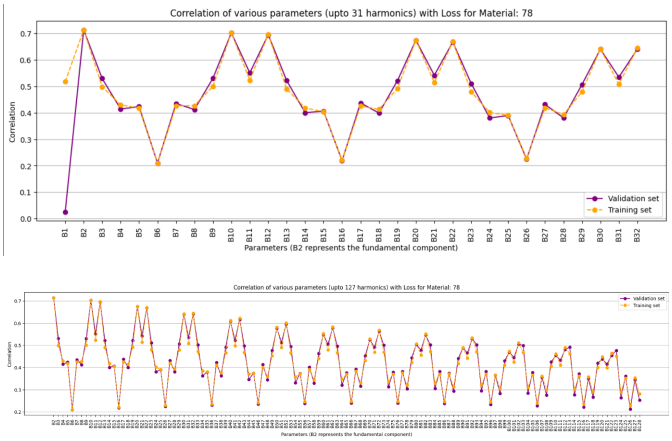

Fig. 3. Correlation plot of Loss vs various parameters for Material 78 with (a) 31 Harmonics and (b) 127 Harmonics

to higher frequencies, the overall volumetric losses slightly decrease. Using $pandas.DataFrame.corr()$, the correlation between B_{max} and Volumetric Loss was found to be 0.759792. In the frequency domain, Figure 3 shows the correlation between various harmonics and loss. Here, B2 represents fundamental or first harmonic, B3 represents second harmonic and so on. It can be seen that different harmonics have different correlations. From Figure 3(a) it can be seen that the 1st, 9th, 11th, 19th, 21st, 29th, 31st harmonics seem to have a much higher level of correlation and this pattern continues in a descending gradient as seen in Figure 3(b).

C. Model Architecture

The neural network model is a multi-input architecture, strategically tailored to address a specific task. There are four different input layers in the model, and each has a specific function. The first input layer is capable of processing data sequences having a particular shape of (32, 1) since it is set up to handle the waveform information. This arrangement corresponds to the expectation of input sequences with length of 32. To extract any temporal dependencies in this frequency domain, two Long Short-Term Memory (LSTM) layers with 128 units each were employed as seen in Figure 4. These LSTM layers play a pivotal role in capturing intricate patterns within the input. To augment the stability of the training process, batch normalization and dropout mechanisms are incorporated in between these two LSTM layers.

TABLE III
DENSE LAYER INFORMATION

Layer	Type	Neurons	Connected to
A	Dense	1024	LSTM Out
B	Dense	512	A
C	Dense	256	B
D	Dense	128	C
E	Dense	64	D
F	Dense	32	E
G	Dense	16	F
Output	Dense	1	G

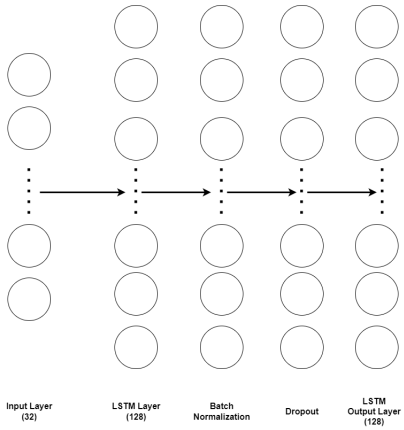

Fig. 4. LSTM Layer Diagaram

The output features obtained from the LSTM layers are subsequently concatenated with the remaining inputs which are frequency, temperature, and classification. This consolidated input is then fed into the subsequent layers of the neural network. The processing of this concatenated input is executed through several dense layers, each equipped with Rectified

Linear Units (ReLU) as activation functions. ReLU activation is a piece-wise linear function which signifies that the node gets activated only if the sum of weighed inputs is greater than 0, else, the node is deactivated, i.e. turned off. ReLU is used here because of its simplicity, general use case and for reduction of vanishing gradient. The information of each of the dense layers is tabulated in Table III.

TABLE IV
MODEL TOTAL PARAMETERS

Material	Model Layer	Total Params	File Size
A	LSTM+ Dense	1033729	3.94 MB
B	LSTM+ Dense	1033729	3.94 MB
C	LSTM+ Dense	1033729	3.94 MB
D	LSTM+ Dense (C to G)	276225	1.09 MB
E	LSTM+ Dense	1033729	3.94 MB

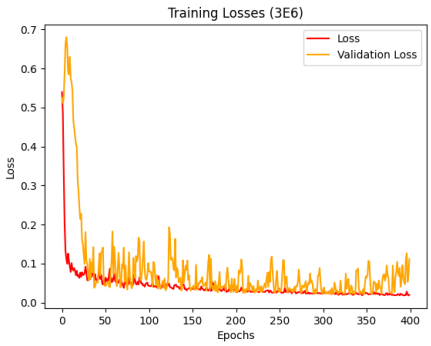

Fig. 5. Training and Validation loss for Material: 3E6

D. Training

Each model was trained for 500+ epochs. During training, $ModelCheckpoint$ under $keras.callbacks$ saves the model after a complete epoch if, at that point, it has a lower value of $validation\ loss$. The plot of training and validation losses, as seen in Figure 5 for the material 3E6, shows that the model learns at a very high rate within the first few epochs, and gradually saturates over time. The noisy nature of Validation loss is due to a smaller batch size which allows the model to learn from each individual example of the training set.

V. RESULTS

A. Error Calculation

Error for each instance is calculated using Equation 2

$$Error(\%) = \frac{|meas - pred|}{meas} * 100\% \quad (2)$$

Here, $meas$ represents the measured loss provided in the dataset, $pred$ represents the prediction of loss outputted by the model. The predictions are outputted in a new list whose values range within the scaling limits. These values are all transformed back using $inverse\_scaler$ to their original range before the error is calculated.

B. Histograms

To analyze the performance of the models, histogram is plotted with percentage relative error as the x-axis and ratio of data points as the y-axis for each material, using $matplotlib$. The plot highlights and labels the overall average error, 95 percentile error and the maximum error. For the known materials, error distribution histogram was plotted for a different validation data-set that was provided separately. However, for the unknown materials, the histogram was plotted for a testing set extracted from the training data using the $train\_test\_split$ function.

C. Testing Results

After training the model using provided data-set for each of the 10 different materials, each model's validity was individually tested using separate validation data sets, each consisting of 5000 randomly sampled data from the original database. The performance of the model for each material can be observed in the histograms as seen in Figures 6. The model was re-fit and tested on 5 unknown materials - A, B, C, D and E. Their validation error histograms can be seen from Figure 7. These sets were provided as a part of the final evaluation of the performance of the model. The training set here varied from thousands of data points for some to a few hundreds of data points for others. In some cases the model was entirely re-trained, whereas in some, a favorable model was re-fit with these additionally provided data points.

VI. MODEL INFORMATION

Table V lists the model details. Each material has its own separate model. The model architecture used for each material is the same, however, certain parameters like epochs, batch size, learning rate etc. are varied for optimal results.

TABLE V
MODEL STATISTICS

No. of input features to model	35
H5 model file size	approx 3 MB
Scaling file size (.joblib file)	1 KB
No. of csv files required before execution	3

VII. CONCLUSION

This research presents a neural-network based approach to predict magnetic core loss. To reduce training complexity, the waveforms are converted to frequency domain by decomposition into harmonics while simultaneously analyzing the waveforms to classify them. The designed neural-network consists of several LSTM layers for feature extraction followed by dense layers with ReLU activation. The model was found to be accurate in predicting core loss for the known materials as shown by the error histograms. Also, the methodology showed promising results in the training data of unknown materials.

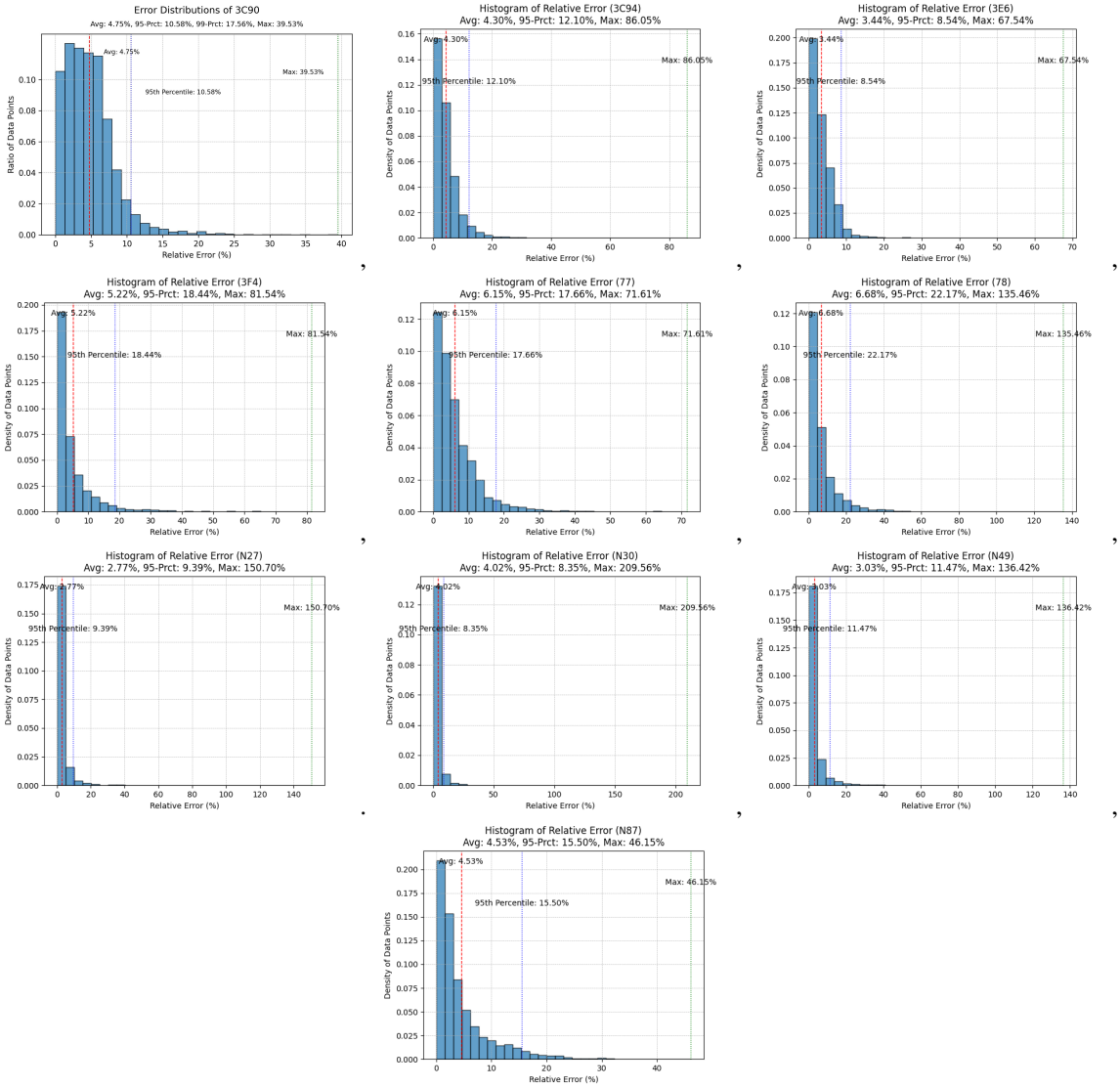

Fig. 6. Error Histogram of 10 different known materials

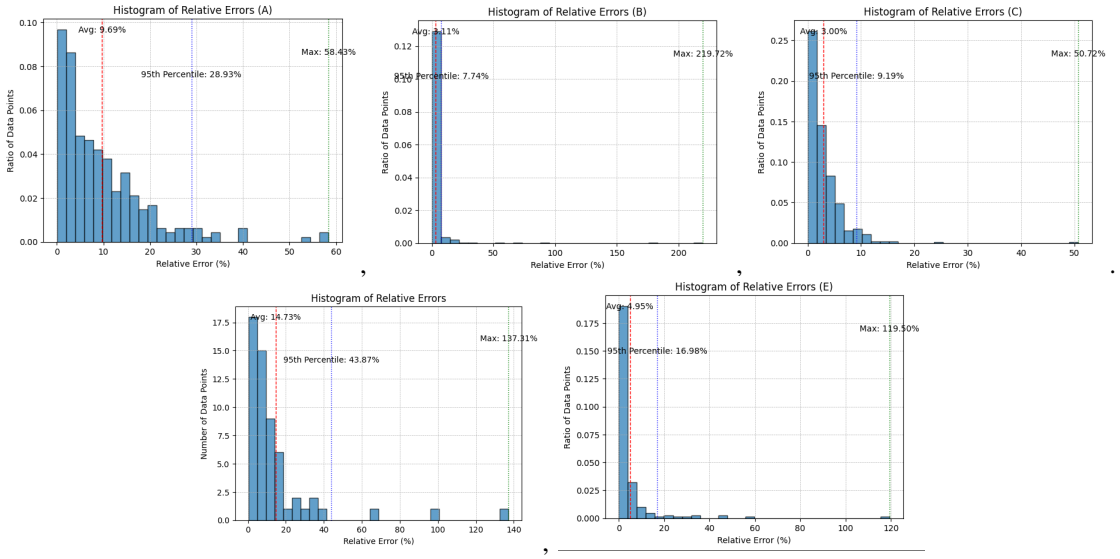

Fig. 7. Validation Error Histogram of 5 different unknown materials

REFERENCES

[1] E. Snelling, *Soft Ferrites Properties and Applications*. Butterworth amp; Co, 1988.

[2] Z. Li, W. Han, Z. Xin, Q. Liu, J. Chen, and P. C. Loh, "A review of magnetic core materials, core loss modeling and measurements in high-power high-frequency transformers," *CPSS Transactions on Power Electronics and Applications*, vol. 7, no. 4, pp. 359–373, 2022.

[3] D. Serrano, H. Li, S. Wang, T. Guillod, M. Luo, V. Bansal, N. K. Jha, Y. Chen, C. R. Sullivan, and M. Chen, "Why magnet: Quantifying the complexity of modeling power magnetic material characteristics," *IEEE Transactions on Power Electronics*, vol. 38, no. 11, pp. 14 292–14 316, 2023.

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

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

[6] S. Zhao, F. Blaabjerg, and H. Wang, "An overview of artificial intelligence applications for power electronics," *IEEE Transactions on Power Electronics*, vol. 36, no. 4, pp. 4633–4658, 2021.

[7] D. Serrano, H. Li, T. Guillod, S. Wang, M. Luo, C. R. Sullivan, and M. Chen, "Neural network as datasheet: Modeling b-h loops of power magnetics with sequence-to-sequence lstm encoder-decoder architecture," in *2022 IEEE 23rd Workshop on Control and Modeling for Power Electronics (COMPEL)*, 2022, pp. 1–8.

[8] X. Shen, H. Wouters, and W. Martinez, "Deep neural network for magnetic core loss estimation using the magnet experimental database," in *2022 24th European Conference on Power Electronics and Applications (EPE'22 ECCE Europe)*, 2022, pp. 1–8.

[9] H. Li, D. Serrano, S. Wang, T. Guillod, M. Luo, and M. Chen, "Predicting the b-h loops of power magnetics with transformer-based encoder-projector-decoder neural network architecture," in *2023 IEEE Applied Power Electronics Conference and Exposition (APEC)*, 2023, pp. 1543–1550.

[10] R. N. Bracewell and R. N. Bracewell, *The Fourier transform and its applications*. McGraw-Hill New York, 1986, vol. 31999.

[11] P. D. Welch, "The use of fast fourier transform for the estimation of power spectra: A method based on time averaging over short, modified periodograms," *IEEE Transactions on Audio and Electroacoustics*, vol. 15, pp. 70–73, 1967. [Online]. Available: https://api.semanticscholar.org/CorpusID:13900622