electronics

MDPI

Article

Machine Learning Approaches for Inverse Problems and Optimal Design in Electromagnetism

Alessandro Formisano [1,*] and Mauro Tucci [2]

[1] Department of Engineering, Università della Campania «Luigi Vanvitelli», 81031 Aversa, Italy
[2] Department of Energy, Systems, Territory and Construction Engineering (DESTEC), Università di Pisa, 56122 Pisa, Italy; mauro.tucci@unipi.it
* Correspondence: alessandro.formisano@unicampania.it

Abstract: The spread of high-performance personal computers, frequently equipped with powerful Graphic Processing Units (GPUs), has raised interest in a set of techniques that are able to extract models of electromagnetic phenomena (and devices) directly from available examples of desired behavior. Such approaches are collectively referred to as Machine Learning (ML). A typical representative ML approach is the so-called "Neural Network" (NN). Using such data-driven models allows the evaluation of the output in a much shorter time when a theoretical model is available, or allows the prediction of the behavior of the systems and devices when no theoretical model is available. With reference to a simple yet representative benchmark electromagnetic problem, some of the possibilities and pitfalls of the use of NNs for the interpretation of measurements (inverse problem) or to obtain required measurements (optimal design problem) are discussed. The investigated aspects include the choice of NN model, the generation of the dataset(s), and the selection of hyper-parameters (hidden layers, training paradigm). Finally, the capabilities in the handling of ill-posed problems are critically revised.

Keywords: machine learning; magnetic field analysis; optimal design; inverse problems

Citation: Formisano, A.; Tucci, M. Machine Learning Approaches for Inverse Problems and Optimal Design in Electromagnetism. *Electronics* **2024**, *13*, 1167. https://doi.org/10.3390/electronics13071167

Academic Editors: Francisco A. Gómez Vela, Miguel García-Torres and Aurelio López-Fernández

Received: 6 February 2024
Revised: 11 March 2024
Accepted: 19 March 2024
Published: 22 March 2024

1. Introduction

Computerized analysis is frequently used to recover field sources or device structures from measurements using numerical computations [1,2]. This process implies the repeated resolution of an electromagnetic problem under different trial values of inputs (either sources or system parameters). The presence of measurement noise, the consideration of manufacturing and assembly tolerances, the inclusion of ferromagnetic materials, or the consideration of complex, three-dimensional geometries add relevant computational efforts to the process. Several measures have been proposed to simplify or speed up the analysis, basically trading off accuracy in the evaluation of the distance of trial data from the actual measurements with promptness.

Alternatively, Machine Learning (ML) and Deep Learning (DL) models can be used for the straightforward resolution of inverse or optimal design problems. This can be done by training the NN to build an (approximated) relationship between the desired output (e.g., sensor readings) and the trial values of the degrees of freedom (e.g., radii or currents of coils, when considering magnets), starting from available examples of the desired output [3–7]. Examples can be obtained by solving a reduced set of instances for the computationally demanding problem, or even be extracted from experimental data.

Note that the construction of an approximate model from available experimental data could reveal the sole possibility in cases where a theoretical problem formulation is not available. This could happen either when a model based on the laws of physics is not available or is too complex to be reduced to a set of equations manageable in the due course of an iterative process.

In addition, the resulting problem, either the inverse one or the design one, shows an ill-posed nature, being prone to multiple solutions [8]. This aspect has been counteracted in many ways, and some precautions must be taken also when using data-driven models.

In previous works, authors have proposed several ML models both for solving the direct [9] model and the inverse model [10], focusing attention on optimization through the use of the direct model and highlighting the difficulty behind the inverse models.

Data-driven models can be obtained following different approaches, including classical statistical regressions or the more modern Deep Neural Networks. Each approach presents advantages and drawbacks which must be comparatively assessed. In addition, the hyper-parameters shaping the approach (e.g., the number of hidden neurons in the NN case) must be carefully chosen to obtain the best balance between promptness and accuracy.

This work presents a detailed analysis of the learning process for a set of NN models of a benchmark inverse problem. The data used to train the model are generated by the FEM or analytical formulations.

The main aspects investigated in this work include the following:

- Training and testing ML approaches to solve direct and indirect electromagnetic problems;
- Selection of the ML model;
- Selection of the model hyper-parameters;
- Dataset generation;
- Regularization approaches;
- How machine learning treats ill-posed inverse problems.

In the following, with reference to a simple yet representative benchmark problem, we will compare the standard *Shallow* Neural Networks (SNNs) [11,12] with a more recent model, namely the Convolutional Neural Network (CNN) [13,14]. For the sake of comparison, additionally a support vector machine, as an example of a regression model, will be introduced.

2. The Benchmark Problem

To compare different approaches, we adopt here the benchmark problem TEAM 35 [15]. A multi-turn air-cored coil is considered. The coil is composed of $n_t = 20$ independent turns. The width of each turn is w = 15 mm and the height is h = 10 mm. Hollow turns are assumed to allow for water circulation. In the following, we will assume the symmetric current distribution among the uppermost 10 coils and the lowermost ones. Consequently, only half of the model is needed to compute the field (see Figure 1a). For evaluating the field, a two-dimensional controlled region is considered, (delimited by curves $S_1, \ldots S_3$ in Figure 1a). The two components B_r and B_z along axes r and z of the flux density field **B** are sampled on a grid evenly spaced in a square region with a side length of 60 mm, which is denoted as the Region Of Interest (ROI). Underlying sensors could include Hall Probes.

To train the shallow neural network and the support vector regressor, we first consider a grid of $n_p = 10 \times 10$ field points, taking the values along the boundaries S1, ... S3 and the line γ, while for the training of the convolutional neural network, we consider a grid of $n_p = 20 \times 20$ field points, taking all the internal values of the square region of interest.

All magnetic analyses that are required to generate the data sets have been performed using an axially symmetric Finite Element (FE) model, with ~7000 quadratic elements. Vanishing flux conditions at the infinite have been imposed using inverted elements. An example of the field map is shown in Figure 1b.

In the TEAM 35 version considered here, the aim of the inverse problem is to identify the radii and the currents in the coils to generate a prescribed flux density map **B**(r,z). In the original formulation, the map is uniform, with $B = B_0\hat{z}$, within a region adjacent to the symmetry plane z = 0 (lines S_1, S_2 and S_3 in Figure 1), and with an amplitude as small as possible along an external segment of the symmetry axis (line γ in Figure 1). To evaluate the field uniformity in the inner ROI, the magnitude of **B** is "measured" in $N_p = 30$ field points, evenly spaced on the boundary of the ROI. On the other hand, in order to guarantee the minimum field amplitude on γ, the field is "measured" on

N_k = 10 points along γ. In particular, the set of measurements includes 60 real values, which correspond to 10 B_r and 10 B_z measurements evenly spaced along lines S_1 and S_2, respectively, 10 B_z values measured along line S_3, and 10 B_z values along line γ. In this paper, we maintain a similar formulation of the inverse problem, where we aim to identify the radii and the currents corresponding to the randomly generated flux density measured in the measurement's points.

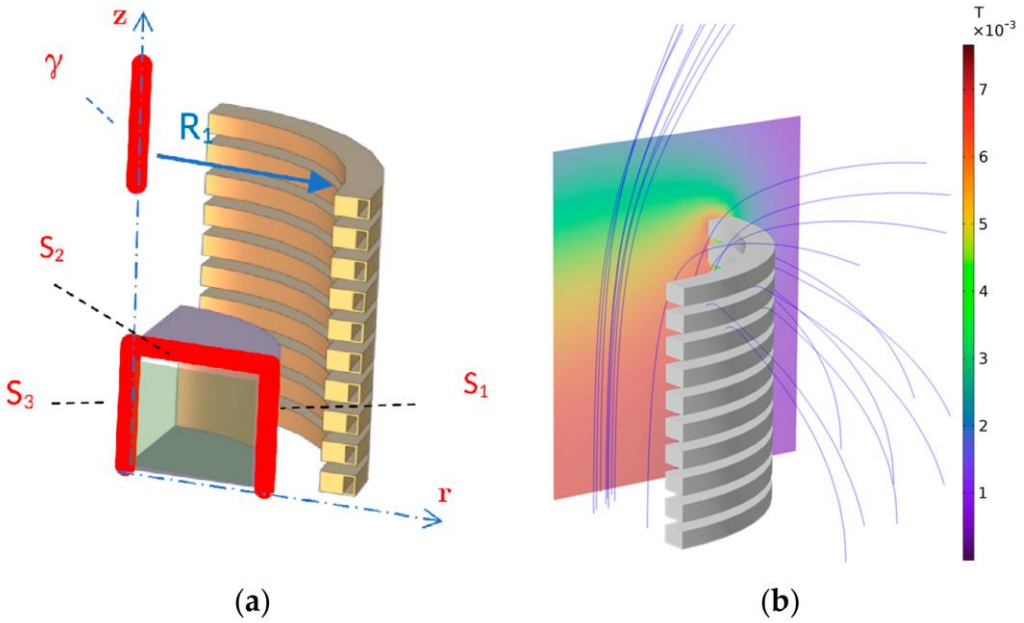


Figure 1. A sketch of the TEAM 35 benchmark problem: computing the magnetic field components along the red lines generated by a set of coils with radii R1, R2, . . . R10. (**a**) the problem geometry, (**b**) an example of the field map.

Figure 2 shows the main characteristics of the variables involved in the problem. Note that when the radii of all coils are known, the relationship between the currents and flux density values is linear. The matrix mapping the current in each coil onto each single measurement is known as the *Lead Field Matrix* (LFM). On the other hand, when the radii are among the degrees of freedom, the inverse problem becomes non-linear, and the LFM must be, in principle, re-assembled for each trial configuration. One of the advantages of the ML approaches is that the reassembly of the LFM is not required, as the NN directly extracts from data the relationship between field values (measurements) and the degrees of freedom (radii and currents).

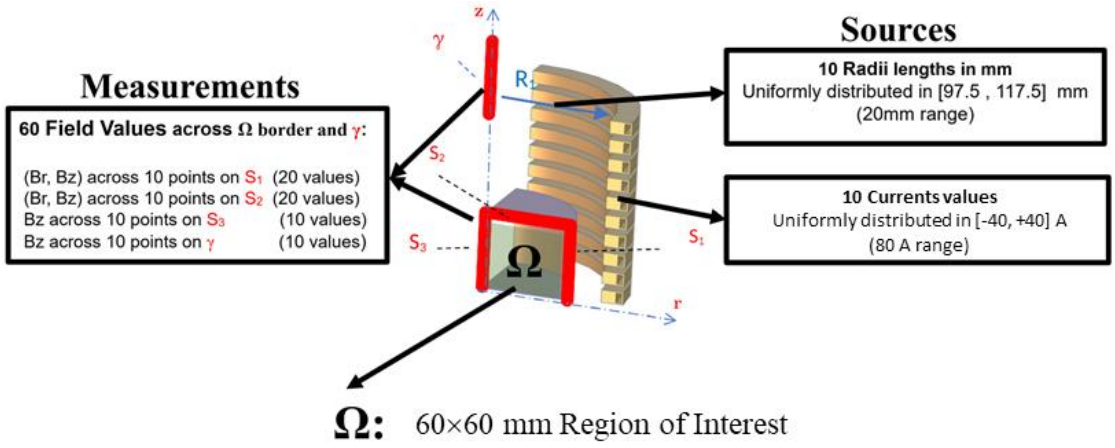


Figure 2. Characteristics of the sources and of the measurements of the benchmark problem.

More details on the benchmark problem geometry can be found in [15].

3. Considered Machine Learning Models

The machine learning models analyzed in this work will be used in different modalities:

- a "Direct Problem" (DP), where examples of the radii/currents set and of the corresponding flux density values are used to create a model able to generate the target field map, hence replicating the LFM from the radii. The direct model is then used within traditional optimization or inverse problem resolution algorithms;
- a first type of "Inverse Problem" (IP#1), where measurements (inputs) *and* currents (outputs) are used to create a model of the underlying linear map. In this case, the radii are assumed known, and the linear model is related to the (pseudo-)inverse of the LFM.
- a second type of "Inverse Problem" (IP#2), where the currents are known and provided as the inputs together with the measurements, while the outputs are the radii;
- finally, a third type (IP#3), where both the currents and radii (the outputs) must be recovered from field measurements, which are the only inputs in this case.

Figure 3 shows the input and output of each defined problem. It is worth pointing out that the above-defined problems exhibit non-linear behavior as soon as the radii are variable. In fact, for fixed radii, the relationship between the currents and the field values is linear, but with varying radii the problem is non-linear, and as such, inherently difficult. To the authors' knowledge, this work is the first attempt to investigate and assess the performance of various machine learning techniques on such a non-linear, low frequency, electromagnetic inverse problem, taking into account all the possible relations between the involved geometrical and electromagnetic variables, and giving to the reader detailed guidelines to approach and interpret inverse electromagnetic problems.

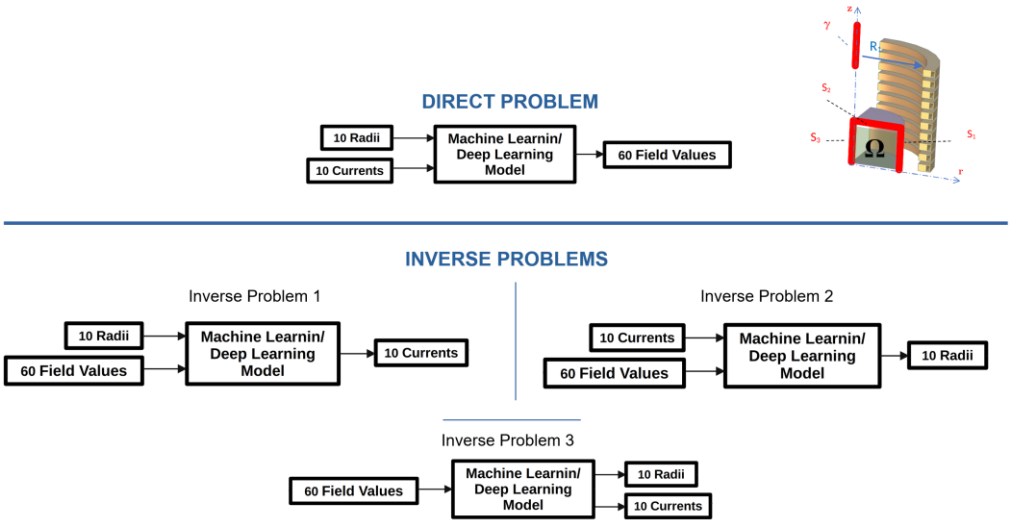


Figure 3. Definition of the direct and inverse problems.

As anticipated, different ML approaches are available to create data-driven behavioral models. In this study, we have considered the following ones:

- Shallow Neural Networks (SNNs) [11,12]. This is the standard approach comprising an artificial neural network with a single hidden layer with logistic activation functions. In the DP, N_i input neurons (corresponding to the radii), N_h hidden neurons and N_o output neurons (the elements of the LFM) are trained to provide the output. While $N_i = 10$, the LFM is represented by a reduced number of real numbers, corresponding to the main components in a PCA based on the correlation analysis of the measurements. The full matrix is then recovered by exploiting the components identified in the PCA. In the inverse problems IP#1, IP#2 and IP#3, the SNN is used

as a straightforward solver, and the meaning of output neurons depends on the type of considered problem. This point will be further discussed below. The number N_h of hidden neurons is varied to assess model capabilities. The different SNNs are trained using the Levenberg–Marquardt Bayesian Regularization approach [16], which minimizes the weights together with the discrepancy in the data. Early topping is performed by means of a worsening performance on a validation set.

- Convolutional Networks (CNNs) [13,14]. This class of NN will be considered for IP#3 only. A higher number of hidden layers is present here, and the network can be named "deep". The inner layers are classified as "convolutional" and "pooling", with different associated actions in the data. The activation functions in this scheme are the "ReLU" functions. This model has an intrinsic capability of building a reduced-order inner model, which can be exploited to cope with the highly correlated nature of the input, represented by the flux density map in the ROI (region Ω in Figure 2). The ADAM algorithm is used to train the CNN.
- In order to compare ML with more traditional statistical regression approaches, we have also considered Support-Vector Regression (SVR) [17,18] for the inverse problems. The SVR training algorithm builds a linear model in a higher dimensional space, exploiting the so-called "kernel trick" by minimizing a quadratic objective function which is a combination of the Euclidean norm of the weights of the linear model and the sum of the so-called slack variables, which represent a threshold of the maximum absolute deviation between the predicted and target values. The LIBSVM implementation of the SVR has been employed.

The hyper-parameters of the SNN and SVR models were selected using an exhaustive K-Fold Grid Search Cross Validation approach, using K = 10 and defining adequate ranges for the optimized hyper-parameters. The CNN architecture was selected using a trial-and-error heuristic approach, which is common for complex deep learning models.

4. Results

To assess the different models, 26,000 examples were generated using a FEM solver to compute the flux density in the measurement points. Examples have been arranged according to radii values to identify different subsets with the same LFM, and divided into training and test sets, according to the 70/30 rule.

4.1. Direct Problem

In the first study, we trained different SNNs on the DP. The metric adopted here to compare the accuracy of the different predicted currents or radii is the Normalized Mean Absolute Error% (NMAE%).

$$\mathrm{NMAE\%} = 100 \frac{\frac{1}{N_{test}} \sum_{N_{test}} \left| Y_{pred} - Y_{true} \right|}{\max(Y_{true}) - \min(Y_{true})}.$$

The NMAE% is an alternative metric that overcomes the limitations of the more classical Mean Absolute Percentage Error in situations involving data that can be negative or close to zero. The NMAE normalizes the error by dividing it by the range of the actual values, providing a more balanced measure of accuracy.

Figure 4 shows a detailed representation of a randomly selected training pattern, including the geometry of the coils and the intensities of the currents. Figure 5a shows that the training of a SNN with 36 hidden neurons for the direct problem required 10 h of computation in a 12 cores processor, while the convergence was obtained after 4 h. The similar trend of the training and validation errors suggests that the SNN is not overfitting. Figure 5b shows a test set pattern prediction with respect to the target value, indicating that the prediction is qualitatively good. The NMAE% for the test set of the direct problem was 0.15%, indicating that the direct problem can be solved with good accuracy.

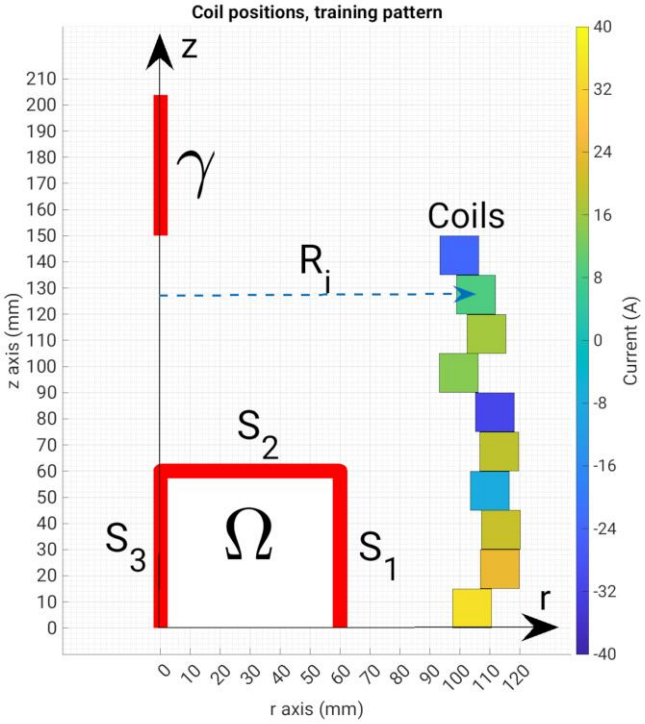
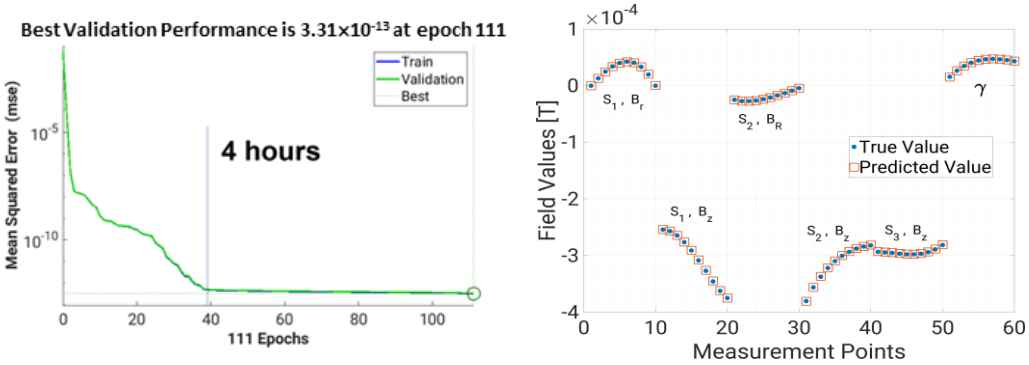


Figure 4. Representation of a random geometry, coils numbering and the ROI.



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

Figure 5. Training error convergence (**a**) and an example of a test field pattern prediction (**b**) of the direct problem solved by means of the SNN.

It is worth pointing out that the complexity of the problem required special attention to the choice of training parameters of the neural network: a second order training function was necessary, using the Levenberg–Marquadt algorithm [11]. This approach updates the vector w of the weights and the biases of the network as follows:

$$w_{k+1} = w_k - \left(J^T J - \mu I\right)^{-1} J^T e$$

where J is the Jacobian vector that contains the first derivatives of the network errors with respect to the weights and biases, and e is a vector of network errors. A small value of the scalar μ results in the Newton method, while a large value leads to gradient descent. The value of μ is adapted during the training: it is decreased after a step that reduces the performance function; otherwise, it is increased. The adaptation is performed by multiplying μ by a decreasing μ_{dec} or by the increasing factor μ_{inc}, respectively. In our experiments we observed that the choice of μ_{dec} and μ_{inc} for this problem is crucial in order to avoid early stopping due to the tendency of μ to diverge, reaching a maximum

allowed value. In particular, we selected $\mu_{dec} = 10^{-2}$ and $\mu_{inc} = 1.001$. In order to obtain the convergence of the training, it was also important to avoid other early stopping mechanisms, such as moniroting reduced performance in the validation set.

4.2. Inverse Problem #1

We then solved the inverse problem type IP#1 by computing the Truncated-Pseudo-Inverse (TSVD) of either the SNN LFM or of the actual LFM, computed using the FEM. Just eight singular modes are used in this investigation. The currents obtained using the two inverse matrices were used to compute the flux density in the measurement points, and the results are compared with the original measurements.

Table 1 reports the test set NMAE% (normalized to 80A) of the 10 currents for the IP#1, solved by means of the SNN.

Table 1. NMAE% for IP#1.

$\mathbf{I_1}$	$\mathbf{I_2}$	$\mathbf{I_3}$	$\mathbf{I_4}$	$\mathbf{I_5}$	$\mathbf{I_6}$	$\mathbf{I_7}$	$\mathbf{I_8}$	$\mathbf{I_9}$	$\mathbf{I_{10}}$
0.91%	0.98%	1.11%	1.12%	1.44%	1.54%	2.22%	2.69%	2.24%	2.21%

From Table 1, it can be observed that the NMAE% increases with the distance of the coil from the sampling curves $S_1 \ldots S_3$. The average NMAE% value is 1.66%. In this case, an SNN with 27 hidden neurons performed best. The average NMAE for SVR is 2.35%. SVR with a Gaussian kernel has three real hyper-parameters and searching them is difficult because a single training requires hours; better results were obtained with an SNN.

The flux density in the "measurement" points corresponding to the different approaches are reported in Figure 6. The scatter plots shown in Figure 7 show that the problem of predicting the currents can be solved by the neural networks accurately, and the dispersion increases with the distance of the coils from the ROI.

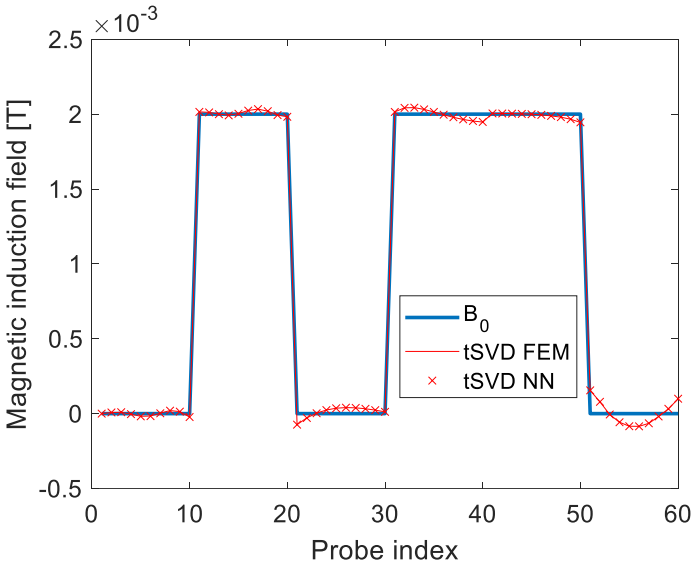


Figure 6. The actual flux density values of the measurement points (blue curve), compared with the values obtained using currents identified using the tSVD of the LFM as generated by the FEM (red continuous line) or as generated by the SNN (red crosses line).

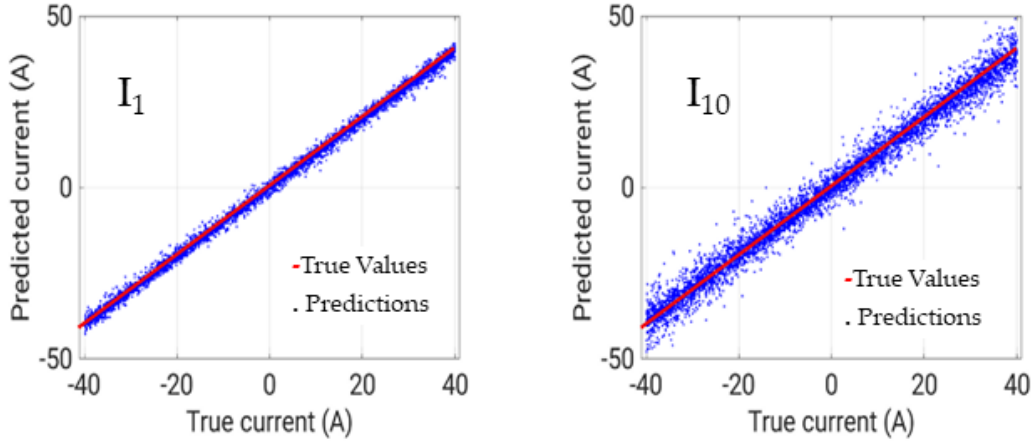


Figure 7. Regression plots of the predicted vs. true currents for the IP#1. The red curve is the exact fit (True Values), while the blue dots are the network prediction.

4.3. Inverse Problem #2

Very similar results are obtained for the inverse problem IP#2, reported in Table 2, again showing an increase in NMAE% with the distance from the measurement region. The averaged value is 10.34% for the SNN and 14.1% for the SVR. In this case, a SNN with 35 hidden neurons performs best.

Table 2. NMAE% for IP#2.

R_1	R_2	R_3	R_4	R_5	R_6	R_7	R_8	R_9	R_{10}
6.75%	7.67%	8.42%	8.71%	10.40%	9.14%	12.94%	13.23%	12.98%	13.14%

Figure 8 evidences that the problem of predicting the radii is more difficult, as expected.

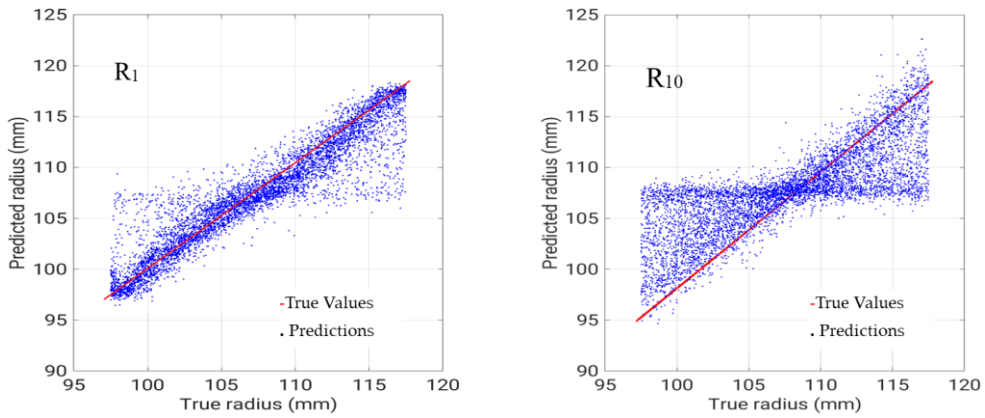


Figure 8. Regression plots of the predicted vs. true radii for the IP#2.

4.4. Inverse Problem #3

Finally, for the inverse problem IP#3, the results for the currents confirm the influence of the distance from the ROI, but with a slight recovery in coils 9 and 10 when the influence of the γ line starts to be relevant. On the other hand, the radii show a completely different behavior (see Table 3).

Table 3. NMAE% for IP#3.

$\mathbf{I_1}$	$\mathbf{I_2}$	$\mathbf{I_3}$	$\mathbf{I_4}$	$\mathbf{I_5}$	$\mathbf{I_6}$	$\mathbf{I_7}$	$\mathbf{I_8}$	$\mathbf{I_9}$	$\mathbf{I_{10}}$
3.53%	4.47%	4.48%	4.51%	4.54%	5.15%	5.85%	7.19%	6.97%	2.21%
$\mathbf{R_1}$	$\mathbf{R_2}$	$\mathbf{R_3}$	$\mathbf{R_4}$	$\mathbf{R_5}$	$\mathbf{R_6}$	$\mathbf{R_7}$	$\mathbf{R_8}$	$\mathbf{R_9}$	$\mathbf{R_{10}}$
24.53%	24.40%	24.62%	26.06%	25.34%	24.61%	25.97%	24.49%	26.04%	24.44%

It is interesting to show the scatter plot of the true vs. predicted radius for coil 1 for all the test cases (Figure 9). Basically, the SNN keeps the estimate of the radius equal to the average value, regardless of the actual value. This is due to the mutual roles of the current and radius in this problem. The expression of the flux density on the axis due to a filamentary coil can be considered as follows:

$$B_{axis} = \frac{\mu_0}{2\pi}\frac{I}{R} \tag{1}$$

where I is the coil current and R its radius; we can conclude that the relevant figure is the ratio of the current and radius. Starting from this, we can hypothesize that the SNN tries to regularize this ill-posed problem by keeping the radius constant and leveraging on the currents only to fit the measurements.

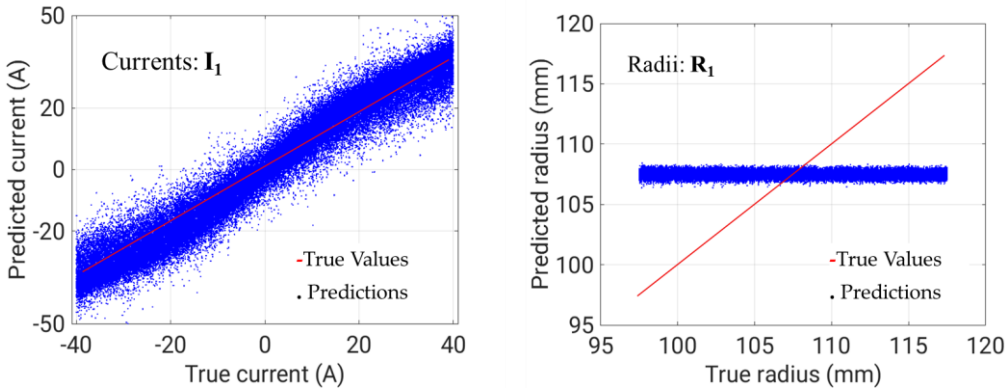


Figure 9. Scatter plot of true and predicted values of current and radius of coil 1.

The above results for the IP#3 were obtained with a SNN with 35 hidden neurons. We addressed the same problem using a CNN, but took as inputs the whole B_r and B_z images inside the ROI, considering a grid of 20 × 20.

The total number of inputs of the CNN is then the whole 2D ROI region 800 (as shown in Figure 10), compared to the 60 inputs along the ROI's border used by the SNN, while the outputs of both methods are the 10 radii values and 10 current values (as shown in Table 3). The architecture of the CNN is shown in Figure 11. The training of the CNN required 20 h, while the training of the SNN and the SVR for the previous problems required 10 h and 4 h, respectively. The results obtained with the CNN are almost identical to the results of the SNN, shown in Table 3 and Figure 6. Considering that the SNN adopts only the boundaries of the ROI as inputs and that the CNN considers the whole 2D domain, this result can be interpreted as a confirmation of the Dirichlet theorem for harmonic functions. The CNN are commonly constructed following empirical rules, requiring experience in DL architectures, and as such, providing good performing architecture for a particular class of problems can be considered an achievement in itself.

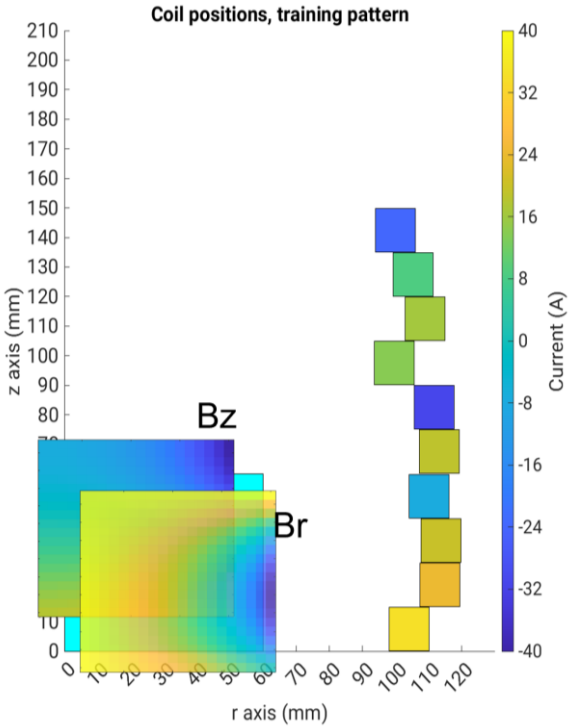


Figure 10. The training pattern for the CNN, depicting the input images of the distribution of Br and Bz inside the region of interest.

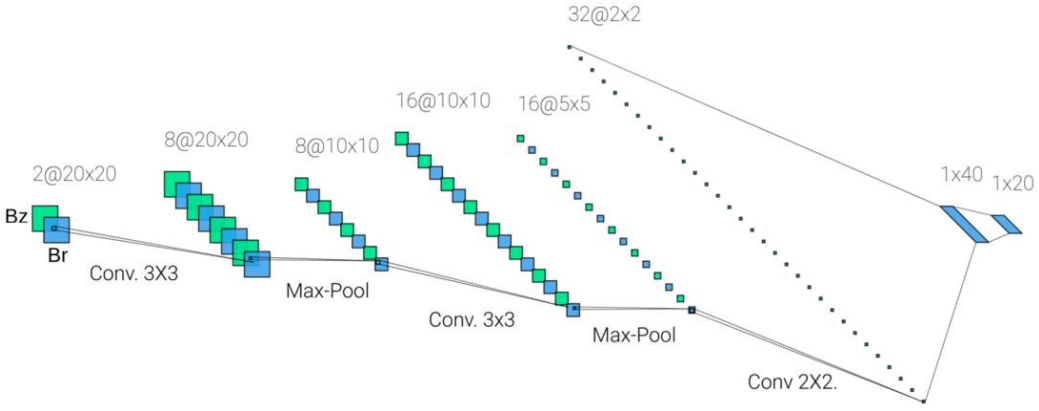


Figure 11. The architecture of the Convolutional Neural Network for inverse problem #3.

Table 4 shows the training and inference time of all the tested methods.

Table 4. Training and inference times, 12 cores CPU amd64 architecture.

Task	SNN	SVR	CNN
Training	10 h	4 h	20 h
Inference	9 ms	11 ms	20 ms

4.5. Inverse Problem #3: Forward Solution of Inverted Patterns

In order to assess the effectiveness of the radii and currents solutions obtained by the neural network for inverse problem IP#3, apart from calculating the errors in Table 3, we used these solutions as the input of the direct solver by means of a Finite Element Method and compared the resulting field distribution with the input field distribution.

As a result, the NMAE% was 4.05%, where the normalization was performed against the mean absolute value of the field inside the ROI.

Figure 12 shows three test examples of the true field values compared to the reconstructed field values using the FEM solution of the geometry obtained by solving the inverse problem IP#3 with the SNN. From these results, it can be stated that even if the error of the predicted radii is large, as can be observed in Figure 6 and Table 3, the combination of the predicted radii and currents represents a good source identification by means of the resulting field that is obtained by applying the direct problem. The leverage of the SNN (and the CNN as well) consisting of applying the non-linearities only to the current estimation appears to be an automatic way for regularizing the difficult inverse problem IP#3.

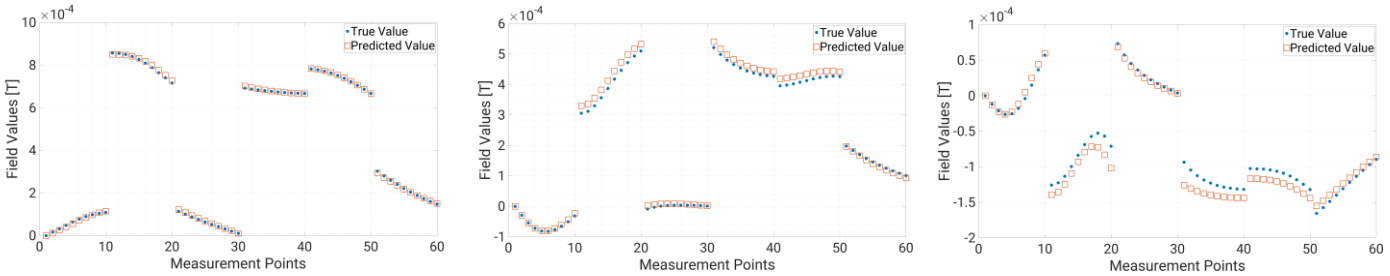


Figure 12. Comparisons between the reconstructed and true values of the field after feeding the FEM direct solver with the neural network solutions of IP#3.

We believe that this result is perhaps the most important finding of this investigation: the reconstructed source, including both geometry and current intensities, is demonstrated to be a valid solution to the problem, and the regularization automatically performed by the NNs leads to feasible and well-behaved solutions. As a remark, the analysis carried out in this work showed that learning the inverse problem directly from the data, and using universal approximators, is an implicit way of generating a regularized model, and we analyzed in detail its performance.

5. Discussion

The availability of different machine learning models for the prompt resolution of electromagnetic problems, as required in the inverse problems from electromagnetic measurements or in the optimization of electromagnetic devices, calls for a reciprocal comparison in terms of the accuracy and promptness of the models, both in the training and operation phases. We proposed here a comparison among a few neural models on a well-known benchmark problem, considered here as an inverse problem.

The correlation among measurements from each sensor location (assessed on 26,000 randomly distributed examples) shows that in the benchmark problem redundancy is present, although this redundancy is well managed by the machine learning and deep learning models, as shown by the results.

In the generation of examples, both fixed radii (linear problem) and varying radii (non-linear problem) were considered. In the linear *direct* problem DP, the reconstruction of the lead field matrix is straightforward, while for the three inverse problems, care must be taken in the choice of the regression algorithm. In the IP#1, the linear case, accurate results are achieved, but when coils with different radii are considered, the network must be re-trained (or a new network considered). In the IP#2 and IP#3 a non-linear inverse problem is faced. Using the available data, satisfactory results were achieved for the IP#2.

In the latter case IP#3, an interpolating model is achieved anyway, but the generalization capability could be probably improved using "noisy" data. The neural network approach to linear inverse problems also requires a careful choice of learning strategy. As a remark, an accurately selected and trained SNN was shown to be a powerful model to solve the inverse problems, both linear and non-linear. The results obtained with the SNN and CNN for the IP#3 show that a similar regularization was performed by two different models to approach the problem of infinite solutions in the inverse problem. Even if the radii and

currents' solutions were strongly regularized, they possessed both feasibilities (they were inside the design bounds), and they thus gave good forward solutions when used as the input of the direct problem. These results might be imputed more into the dataset itself than to the ML approach, confirming the results obtained in previous works [13]: the choice of the distribution of design parameters used to generate the training set is crucial for the performance of the ML approaches that are, of course, data-driven.

Another quite remarkable result, which is new in the literature, is that the inversion of the electromagnetic problem gives the same results when using as input the field distribution in the boundary of the domain and the field distribution in the whole domain. The distribution inside the domain is, in theory, dependent on the boundary conditions, and thus is not a complete surprise from a mathematical point of view, but it is not automatically expected that a neural network is able to incorporate this knowledge based on learning from data. The results obtained for the IP#3 solved by means of the SNN, which uses the boundary of the ROI, and the CNN, which uses the whole ROI, show that the SNN was able to acquire from the boundary values the same information that the CNN obtained from internal values. These results find confirmation in the recent development of physics-informed neural networks. For this reason, we are still working on the resolution of electromagnetic inverse and optimization problems with physics-informed [19] and generative adversarial (game theory) neural networks [20].

It is worth finally remarking that, if the geometry of the problem involves 10 coils (with varying radii and currents) and axial symmetry, the trained neural networks can solve these geometries both for the direct and indirect problem. In fact, a strength of the proposed approach is that we are not just solving a single geometry, but a class of geometries with varying shapes, as we consider varying radii. If some other geometrical parameters change, such as the number, shape or width of the coils, a new NN shall be trained. In the case of CNNs, future work shall be related to using transfer learning to reduce the time needed to learn new classes of geometries.

Future work will be focused on the tuning of the NN for the inverse problem with a custom training loop using automatic differentiation. The idea is to back-propagate the error of the reconstructed field obtained by feeding the direct solver with the result of the NN.

Author Contributions: Conceptualization, A.F.; methodology, A.F. and M.T.; software, M.T.; formal analysis, A.F.; investigation, A.F. and M.T.; data curation, M.T.; writing—original draft preparation, A.F.; writing—review and editing, M.T. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Data Availability Statement: The article presents all the methods and details required to generate the data analyzed in this work.

Acknowledgments: The authors wish to thank Sami Barmada from Università di Pisa and Paolo di Barba and Maria Evelina Mognaschi for their continuing support and discussions.

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

References

1. Mohammed, O.A.; Lowther, D.A.; Lean, M.H.; Alhalabi, B. On the creation of a generalized design optimization environment for electromagnetic devices. *IEEE Trans. Magn.* **2001**, *37*, 3562–3565. [CrossRef]
2. Massa, A.; Salucci, M. On the design of complex EM devices and systems through the System-by-Design paradigm: A framework for dealing with the computational complexity. *IEEE Trans. Antennas Propag.* **2021**, *70*, 1328–1343. [CrossRef]
3. Li, Y.; Lei, G.; Bramerdorfer, G.; Peng, S.; Sun, X.; Zhu, J. Machine learning for design optimization of electromagnetic devices: Recent developments and future directions. *Appl. Sci.* **2021**, *11*, 1627. [CrossRef]
4. Khan, A.; Lowther, D.A. Machine learning applied to the design and analysis of low frequency electromagnetic devices. In Proceedings of the IEEE 21st International Symposium on Electrical Apparatus & Technologies (SIELA 2020), Bourgas, Bulgaria, 3–6 June 2020; pp. 1–4. [CrossRef]
5. Khan, A.; Ghorbanian, V.; Lowther, D. Deep learning for magnetic field estimation. *IEEE Trans. Magn.* **2019**, *55*, 1–4. [CrossRef]

6. Khan, A.; Mohammadi, M.H.; Ghorbanian, V.; Lowther, D. Efficiency map prediction of motor drives using deep learning. *IEEE Trans. Magn.* **2020**, *56*, 1–4. [CrossRef]
7. Rahman, M.M.; Khan, A.; Lowther, D.; Giannacopoulos, D. Evaluating magnetic fields using deep learning. *COMPEL-Int. J. Comput. Math. Electr. Electron. Eng.* **2023**, *42*, 1113–1130. [CrossRef]
8. Formisano, A.; Martone, R. Different regularization methods for an inverse magnetostatic problem. *Int. J. Appl. Electromagn. Mech.* **2019**, *60*, S49–S62. [CrossRef]
9. Tucci, M.; Barmada, S.; Formisano, A.; Thomopulos, D. A regularized procedure to generate a deep learning model for topology optimization of electromagnetic devices. *Electronics* **2021**, *10*, 2185. [CrossRef]
10. Barmada, S.; Di Barba, P.; Formisano, A.; Mognaschi, M.E.; Tucci, M. Learning-Based Approaches to Current Identification from Magnetic Sensors. *Sensors* **2023**, *23*, 3832. [CrossRef] [PubMed]
11. Haykin, S. *Neural Networks and Learning Machines*, 3rd ed.; Prentice Hall: New York, NY, USA, 2011; ISBN 13-978-8131763773.
12. Aggarwal, C.C. Machine Learning with Shallow Neural Networks. In *Neural Networks and Deep Learning*; Springer: Cham, Switzerland, 2018; pp. 53–104, ISBN 978-3-319-94463-0.
13. Goodfellow, I.; Bengio, Y.; Courville, A. *Deep Learning*; MIT Press: Cambridge, MA, USA, 2016; ISBN 13-9780262035613.
14. Aggarwal, C.C. Convolutional Neural Networks. In *Neural Networks and Deep Learning*; Springer: Cham, Switzerland, 2018; pp. 315–371, ISBN 978-3-319-94463-0.
15. Alotto, P.; Di Barba, P.; Formisano, A.; Lozito, G.M.; Martone, R.; Mognaschi, M.E.; Repetto, M.; Salvini, A.; Savini, A. Synthesizing sources in magnetics: A benchmark problem. *COMPEL-Int. J. Comput. Math. Electr. Electron. Eng.* **2021**, *40*, 1084–1103. [CrossRef]
16. Foresee, F.D.; Hagan, M.T. Gauss-Newton approximation to Bayesian learning. In Proceedings of the International Joint Conference on Neural Networks (IJCNN 1997), Houston, TX, USA, 12 June 1997; pp. 1930–1935.
17. Cortes, C.; Vapnik, V. Support-vector networks. *Mach. Learn.* **1995**, *20*, 273–297. [CrossRef]
18. Smola, A.J.; Schölkopf, B. A tutorial on support vector regression. *Stat. Comput.* **2004**, *14*, 199–222. [CrossRef]
19. Khan, A.; Lowther, D.A. Physics informed neural networks for electromagnetic analysis. *IEEE Trans. Magn.* **2022**, *58*, 1–4. [CrossRef]
20. Creswell, A.; White, T.; Dumoulin, V.; Arulkumaran, K.; Sengupta, B.; Bharath, A.A. Generative adversarial networks: An overview. *IEEE Signal Process. Mag.* **2018**, *35*, 53–65. [CrossRef]