Performance analysis of Electrical Machines based on Electromagnetic System Characterization using Deep Learning

Vivek Parekh¹,², Dominik Flore², and Sebastian Schöps¹
¹Technical University Darmstadt, Computational Electromagnetics Group, 64289 Darmstadt, Germany
²Robert Bosch GmbH, Engineering, Acquisition, Building Set (PS-EM/EAB), 70442 Stuttgart, Germany

The numerical optimization of an electrical machine entails computationally intensive and time-consuming magneto-static finite element (FE) simulation. Generally, this FE-simulation involves varying input geometry, electrical, and material parameters of an electrical machine. The result of the FE simulation characterizes the electromagnetic behavior of the electrical machine. It usually includes nonlinear iron losses and electromagnetic torque and flux at different time-steps for an electrical cycle at each operating point (varying electrical input phase current and control angle). In this paper, we present a novel data-driven deep learning (DL) approach to approximate the electromagnetic behavior of an electrical machine by predicting intermediate measures that include non-linear iron losses, a non-negligible fraction (1/6 of a whole electrical period) of the electromagnetic torque and flux at different time-steps for each operating point. The remaining time-steps of the electromagnetic flux and torque for an electrical cycle are estimated by exploiting the magnetic state symmetry of the electrical machine. Then these calculations, along with the system parameters, are fed as input to the physics-based analytical models to estimate characteristic maps and key performance indicators (KPIs) such as material cost, maximum torque, power, torque ripple, etc. The key idea is to train the proposed multi-branch deep neural network (DNN) step by step on a large volume of stored FE data in a supervised manner. Preliminary results exhibit that the predictions of intermediate measures and the subsequent computations of KPIs are close to the ground truth for a new machine design in the input design space. This hybrid approach yields flexibility in the simulation process, e.g., the value of the system parameters can be changed without performing a new FE simulation. In the end, the quantitative analysis validates that the hybrid approach is more accurate than the existing DNN-based direct prediction of KPIs, which avoids electromagnetic calculations.

Index Terms—finite element, flux, deep neural network, torque

I. INTRODUCTION

Motivation

The electrical machine has numerous real-world applications, ranging from home appliances to the automotive industry. Permanent magnet synchronous machine (PMSM), a type of electrical machine, has gained popularity in recent years due to its various advantages, such as high efficiency, greater power density, and higher torque-current ratio. The use of expensive materials such as rare earth magnets (neodymium, dysprosium, or terbium) is a significant cost operator in the PMSM manufacturing process. As a result, it necessitates numerical optimization of an electrical machine in a high-dimensional input design space [1]. The numerical optimization yields optimal design specifications that propel the cost-effective production of an electrical machine design, and therefore, it is a vital part of the whole design process. The finite element method (FEM) based electromagnetic calculations of an electrical machine are at the heart of numerical optimization. In the automotive industry, magneto-static FEM-based optimization has been used for decades to reduce the number of physical prototypes and to optimize the active parts of the electrical machine (stator, rotor, magnets, winding) in the early design phase while lowering costs. The FEM-based simulation enables us to understand and quantify the physical phenomenon of electromagnetic behavior of an electrical machine. It is an intensively time-consuming and computationally expensive process, so there is a great need for a flexible, accurate, and computationally cost-saving approach.

This research builds on the findings of paper [1]. It demonstrates how a deep learning (DL) based meta-models predict cross-domain KPIs effectively using different PMSM representations in a large input design space. As shown in [1], the meta-model trained with parameter-based data has higher prediction accuracy than the image-based representation. The generation of (high-resolution) images and the training of a deep convolutional neural network requires more time and effort than the parameter-based approach. However, the image-based meta-model remains invariant to re-parametrization that does not hold for the parameter-based model. All the meta-models are trained directly on output KPIs using supervised learning. It makes the DL meta-model deterministic to the range of KPIs over which it is trained. During the conventional optimization process, the final single-valued ground truth KPIs are calculated after the magneto-static FE simulation phase using a powerful physics-based post-processing tool. This post-processing tool requires magneto-static FE output, system parameters such as electrical input...
for the inverter (DC voltage, current), desired set points (torque and speed), scaling parameters (stack length), winding parameters (e.g., number of parallel wires, branches), etc., as input to compute the final single-valued KPIs. During the data generation process, the output KPIs are obtained with constant values of these system parameters. Thus, the training of the DL meta-model implicitly depends on one set configuration of system parameters and, therefore, the direct prediction of KPIs becomes deterministic towards this setup. The post-processing tool has tremendous calculation power as it analyzes electrical machine design in detail by computing various performance characteristics and efficiency maps. As a result, many DL meta-models with separate training might be required to evaluate such a wide range of performance indicators. The DL-based optimization approach must be independent of system parameters and the post-processing calculations in order to overcome these limits and boost flexibility and analysis capacity. Therefore, in this paper, we present a novel data-driven DL-based approach to characterize the electromagnetic behavior of an electrical machine. The proposed multi-branch deep neural network (DNN) functions as a meta-model with varying scalar parameters as input to approximate the electromagnetic behavior of the electrical machine. The magneto-static finite element (FE) simulations are performed in high dimensional input space. The results of the simulations are stored in a large database. The aim is to retrieve FE results step-by-step and train multi-branch DNN using supervised learning. We demonstrate how the trained multi-branch DNN can be used to lower the computational overhead incurred by repeated FE simulations. The multi-branch DNN is designed to handle a large number of different outputs. The trained multi-branch DNN predicts part of magneto-static FE simulation referred to as intermediate measures that include non-linear iron losses (hysteresis and eddy current losses), a non-negligible fraction (60° of one electrical cycle) of electromagnetic torque, and flux values at different time-steps for each operating point. The remaining time steps are being calculated by utilizing the magnetic state symmetry of the PMSM, which automatically leads to a significant reduction of the computational burden. The calculated FE result, along with the system parameters, is then embedded into physics-based analytical models to calculate characteristic maps and cross-domain key performance indicators (KPIs), i.e., maximum torque, power, cost, etc. Finally, we quantitatively compare this hybrid method to a direct DL approach described in [1], where a deep neural network (DNN) is trained solely on varying scalar parameters and serves as a meta-model for predicting the KPIs.

The article is organized as follows: Section II briefly discusses the state of the art applications of DL in the field of electrical machine. Section III explains problem formulation and data set specifics. Section IV presents the methodology, network architecture, and training settings. Section V provides quantitative analysis of results which is followed by the conclusion.

II. STATE OF THE ART

In recent years, deep learning applications in the field of electrical machine have grown. The DNN is employed as a meta-model at various design stages because of its numerous advantages such as scalability for high dimensional data, handling of big data, easy parallelization over GPUs, incorporation of real-world data, automatic feature extraction, etc. An overview of different DL-based intelligent fault diagnosis techniques for rotating machinery has been given in [2]. This paper [3] describes a DL-based approximation of electromagnetic torque for PMSM. In [1], it is demonstrated how to predict a large number of cross-domain KPIs in high dimensional input space. In another article, the DNN is used as a meta-model for predicting objectives obtained by FE output in the design phase for the optimization of flux switching machine to lower the computational burden [4]. In [5], it is shown how convolutional neural network (CNN) based methods employed to accelerate the multi-objective topology optimization. Another work [6] demonstrates how a deep learning-assisted approach is used to predict efficiency maps of electric drives. The DNN-based two-step optimization method is investigated in [7] to speed up the design process by inferring torque-related performance. The article [8] discusses machine learning-based methods for approximating different performance maps, as well as the possibility of transferring knowledge through trained DNN for other electrical machine topologies via transfer learning concept. The study in [9] provides an overview of recent developments and future directions in design optimization methods for electromagnetic devices, with a focus on machine learning methods. The application of CNN to approximate performance characteristics of electric machines for multidimensional outputs is probed in [10]. The use of DL-based meta-models for topology optimization was studied in order to reduce the overall computational time of the optimization procedure [11]-[13]. In the recent past, the article [14] explored a variational autoencoder-based DL approach for topology optimization of electromagnetic devices.

III. PROBLEM FORMULATION AND DATA SET
A. Problem formulation

In this work, magneto-static FE simulation for each electrical machine design requires high-dimensional varying scalar parameter-based input (geometry, electrical excitation, and material) and real-valued multiple-outputs which primarily comprise of non-linear iron losses ($\mathcal{L}$), electromagnetic flux associated with all three coils ($\Psi_1, \Psi_2, \Psi_3$) and electromagnetic torque ($T$). Let us consider simulation database $\mathcal{D}$ of N electrical machine designs for which every scalar input
vector is generated by $d$-dimensional random variable $\mathbf{p}$ and output targets by $m$-dimensional random variable $\mathbf{y}$ that contains significant part of FE simulation result. The data set $\mathcal{D}$ can be written as,

$$\mathcal{D} := \{ (p^{(1)}, y^{(1)}), ..., (p^{(N)}, y^{(N)}) \} \text{ for all } N \}.$$ (1)

Thus, for each electrical machine design, the non-negligible fraction of FE simulation result is characterized by $d \times 1$ sized input vector $p^{(i)} := (p_1^{(i)}, ..., p_d^{(i)})$, and $m \times 1$ sized target vector $y^{(i)} := (y_1^{(i)}, ..., y_m^{(i)})$ with $i \in \{1,...,d\}$, $j \in \{1,...,m\}$ and $l \in \{1,...,N\}$. The goal is to train a multi-target regression model from $\mathcal{D}$ consisting of finding a function $\mathcal{M}_{FE}$ that approximates substantial part of FE simulation result for each machine design instance given by the vector $\mathbf{p}$ and a vector $\mathbf{y}$ of $m$ target values:

$$\mathcal{M}_{FE} : \mathcal{P} \rightarrow \mathcal{Y}$$

$$\mathbf{p} = (p_1, ..., p_d) \mapsto \mathbf{y} = (y_1, ..., y_m) \quad (2)$$

where $\mathcal{P} \subset \mathbb{R}^d$ and $\mathcal{Y} \subset \mathbb{R}^m$ denote the sample spaces for predictive variable $\mathbf{p}$ and target variable $\mathbf{y}$, respectively. The learned function $\mathcal{M}_{FE}(\cdot)$ will be used afterwards as a FE meta-model to simultaneously predict the target values for new unseen machine designs.

**B. Dataset specifics**

In this study, the simulation dataset is generated for the double-V topology of PMSM. The procedure for calculating KPIs is detailed in section 2A of [1]. A total of $N_{EM} = 44877$ machine designs have been simulated for 35 varying input parameters (geometry and material). Each machine design undergoes FE calculation for 37 operating points. The operating point is counted as a variable electrical excitation input for the electrical machine. It is primarily a combination of input phase current ($I_{phase}$) and control angle ($\alpha$) associated with it. In this scenario, the control angle ($\alpha$) is the electrical phase angle formed by the phase current and the pole wheel voltage (induced voltage in the rotor). As a result, $N_D = 37 \times N_{EM} = 1660449$ will be the actual number of samples (dataset) on which the meta-model will be trained and tested. The result of each operating point magneto-static FE calculation is the non-linear iron losses ($V_{fe}$), and electromagnetic torque ($T$) and flux linked with three coils ($\Psi_{coil1}, \Psi_{coil2}, \Psi_{coil3}$) for one electrical period. The non-linear iron losses include two physical losses - hysteresis losses ($V_{fe, hyst}$) and eddy current losses ($V_{fe, eddy}$). The analytical iron-loss model given by Jordan [15], [16] is used to compute these physical losses for rotor and stator. This dataset is generated for the real-world industrial project under the following conditions: 1) each machine design adheres to input geometry symmetry 2) three-phase stator windings are shifted by 120° and fed by temporally shifted (120°) three-phase identical currents for input electrical excitation 3) integer-slot winding (means slots per pole per phase remains integer number) 4) additional harmonic losses are ignored.

The first three conditions imply that the magnetic state of the electrical machine will remain unchanged after a 60° electrical rotation. It allows us to exploit the magnetic state symmetry of the electrical machine. It means that each magneto-static FE simulation is run for only $\frac{1}{6}$ of electrical period, and the remainder is determined by extrapolating the result for the whole input electrical cycle. This process has already to be very effective in reducing the computational burden. Therefore, the goal is to train the meta-model to predict as accurately as possible the iron losses ($V_{fe}$), and the non-negligible fraction of torque ($T_{60}$) and flux ($\Psi_{coil1}, \Psi_{coil2}, \Psi_{coil3}$) calculation at each operating point. In this context, this is also termed as intermediate measures prediction. [Figure 1] depicts one representative geometry (full pole cross-section) from the dataset. Out of a total of 35 varying parameters (geometry, material), Table I provides information on 5 important parameters, while Table III includes details on the other input parameters. The plot of one operating point calculation ($\Psi_{coil1}, \Psi_{coil2}, \Psi_{coil3}, T$) of the whole electrical period for the sample electrical machine design from the dataset is shown in [Figure 2].

The distribution of five KPIs and parameters can be seen in [Figure 3].

| Parameter       | Min  | Max  | Unit       |
|-----------------|------|------|------------|
| $p_1$           | 0.50 | 2.0  | [mm]       |
| $p_2$           | 2.23 | 7.0  | [mm]       |
| $p_3$           | 8.0  | 25.0 | [mm]       |
| $p_4$           | 12   | 20   | [mm]       |
| $p_5$           | 159  | 165  | [mm]       |

**Table I: Geometry parameters detail**

| KPIs                | Unit       |
|---------------------|------------|
| $y_1$               | Maximum torque on limit curve [Nm] |
| $y_2$               | Max shaft power [W] |
| $y_3$               | Max shaft power@max speed [W] |
| $y_4$               | Maximum torque ripple on limit curve [Nm] |
| $y_5$               | Material cost [Euro] |
| $y_6$               | Mass of active parts [Kg] |
| $y_7$               | Torque ripple deviation [Nm] |

**Table II: KPIs detail**

| Parameters          | Value | Unit |
|---------------------|-------|------|
| No of pole pair     | 4     |     |
| No of pole pairs    | 4     |     |
| Winding             | Symmetric |
| Motor type          | Double-V topology |
| Maximum input phase voltage | 1336.4 |     |
| Maximum input current ($I_{phase, max}$) | 1336.4 |     |
| Varying input phase current | 37/1336.4 |     |
| Varying control angle | 37/1336.4 |     |

**Table III: Other input parameters detail**
IV. PROCEDURE, NETWORK ARCHITECTURE AND TRAINING

This section first discusses the proposed procedure for calculating KPIs via characterization of electromagnetic system of electrical machine using multi-branch DNN, followed by specifics on the proposed multi-branch DNN architecture and its training.

A. Procedure

The Figure 4 shows block diagrams of three different approaches for the calculation of KPIs. The classical approach is a conventional way of computing KPIs and is widely used in industry. In this method, as shown in the diagram, input parameters (material, geometry, and electrical excitation) are fed as an input to magneto-static FE simulation, and then the result of it, along with system parameters, are embedded into physics-based in-house post-processing tools to estimate final KPIs. The true functions for computing FE results and KPIs, respectively. Iron loss ($V_{fe}$) is a vector with four components ($V_{fe}^{hyst}$, $V_{fe}^{eddy}$, $V_{fe}^{hyst}$, $V_{fe}^{eddy}$) that denotes hysteresis and eddy current losses for stator and rotor. Similarly, $\Psi$ is a vector of three variables ($\Psi_{coil1}$, $\Psi_{coil2}$, $\Psi_{coil3}$) that describe the flux-linkages of all three coils. The system parameters ($S$) are required input parameters (e.g., scaling, geometry, winding parameters, etc.) for post-processing to estimate KPIs.

In the hybrid approach as shown in the Figure 4, we use multi-branch DNN as a meta-model to approximate true function ($M_{FE_{true}}$) for FE calculation. So the same way as Equation 3 for the hybrid approach can be written as,

\[
(\hat{V}_{fe}, \hat{\Psi}, \hat{T}) := M_{\theta}(p_1, ..., p_d, I_{phase}, \alpha)
\]

\[
(\hat{y}_1, ..., \hat{y}_m) := K_{true}(\hat{V}_{fe}, \hat{\Psi}, \hat{T}, S)
\]

where, $K_{true}(\cdot)$ and $M_{FE_{true}}(\cdot)$ are the actual functions for computing FE results and KPIs, respectively. Iron loss ($V_{fe}$) is a vector with four components ($V_{fe}^{hyst}$, $V_{fe}^{eddy}$, $V_{fe}^{hyst}$, $V_{fe}^{eddy}$) that denotes hysteresis and eddy current losses for stator and rotor. Similarly, $\Psi$ is a vector of three variables ($\Psi_{coil1}$, $\Psi_{coil2}$, $\Psi_{coil3}$) that describe the flux-linkages of all three coils. The system parameters ($S$) are required input parameters (e.g., scaling, geometry, winding parameters, etc.) for post-processing to estimate KPIs.

The $\hat{y}_1, ..., \hat{y}_m$ are predicted KPIs. $\theta$ presents model parameters for trained multi-branch DNN. The goal of multi-branch DNN training is to optimize model parameters $\theta$ by minimizing the difference between ground truth and prediction for each sample. The training loss function ($\ell_1$-norm) can be written as follows:
\[ \mathcal{L}(\theta) := \| \mathcal{M}_{FE, true}^{(j)} - \mathcal{M}_{\hat{\theta}}^{(j)} \| \] (5)

Where, \( \mathcal{M}_{FE, true}^{(j)} \) is actual FE calculation and \( \mathcal{M}_{\hat{\theta}}^{(j)} \) predicted FE output for the input sample \( (j) \). In our previous work [1], the direct DL approach is detailed. The idea is to approximate the true KPI function \( (K_{true}(\cdot)) \) via DNN using supervised learning. So the approximated function with DNN can be written as,

\[ (\hat{y}_1, ..., \hat{y}_m) := \hat{K}_\Phi (p_1, ..., p_d) \] (6)

The approximated KPIs function \( \hat{K}_\Phi (\cdot) \) works as a meta model to predict KPIs \( \hat{y}_1, ..., \hat{y}_m \), and \( \Phi \) are network parameters. The training goal of the meta-model remains same as of the multi-branch DNN, therefore, the loss function for the training is described as,

\[ \mathcal{L}(\Phi) := \| \hat{K}_{true}^{(j)}(\cdot) - \hat{K}^{(j)}(\cdot) \| \] (7)

Where, \( \hat{K}_{true}^{(j)}(\cdot) \) is ground truth KPIs and \( \hat{K}^{(j)}(\cdot) \) predicted KPIs for the input design \( (j) \).

B. Network architecture and training

The performance of any deep learning model on a given dataset principally relies on setting the correct values for a combination of hyperparameters. The hyperparameters are set before training, whereas the model parameters, such as weights of DNN, are optimized during training. For better understanding, we can divide hyperparameters into two categories: model hyperparameters and learning hyperparameters. The model hyperparameters are specified as the width and height of different network layers, filter size, stride, type of layer (dense, convolutional, max-pooling, etc.), and so on. The learning hyperparameters include, for example, learning rate, activation functions, optimizer, loss function, batch size, epochs, etc. To extract the maximum performance out of the meta-model, it is paramount to search proper values for a combination of hyperparameters and design a network that can handle high dimensional input and a large number of outputs. Therefore, we propose multi-branch DNN as depicted in Figure 3. At first, this structure was fixed through trial and error by evaluating its performance with roughly ten different configurations, and then we targeted hyperparameters listed in Table IV for optimization. The hyperparameter optimization (HPO) was performed using an unpublished in-house optimization tool that includes implementation of Asynchronous Successive Halving Algorithm (ASHA) algorithm [17] for a multi-objective case. This tool has evaluated 800 different configurations within search space defined in Table IV. The entire HPO process took approximately two days with parallelization across four GPUs. The detailed discussion of the HPO procedure is beyond the scope of this article. The final optimal configuration is given in the last column of Table IV. As shown in Figure 5, input layer has a size of \( 37 \times 1 \) as there are 37 varying input parameters. There are four common layers (CL: 1590 \( \rightarrow \) 1410 \( \rightarrow \) 810 \( \rightarrow \) 210) to exploit correlation among all the output measures. There are total five distinct branches with different number of branch layers (BL). There are two identical branches for torque \( (T) \) and iron loss \( (\hat{\Psi}_r) \) prediction, and three equal size branches for flux \( (\Phi) \) prediction. The network layers for iron loss and torque: 1530 \( \rightarrow \) 1210 \( \rightarrow \) 900 \( \rightarrow \) 880 \( \rightarrow \) 750 \( \rightarrow \) 660 \( \rightarrow \) 610 \( \rightarrow \) 580 \( \rightarrow \) 550 \( \rightarrow \) 530. The branch configuration for other three branches are: 322 \( \rightarrow \) 278 \( \rightarrow \) 240. As stated earlier in the section Section III-B each branch will predict intermediate measures \( (60^\circ \text{ of a whole electrical cycle}) \) for every operating point. In this research, we have a 1-time step = electrical \( 4^\circ \) setup. As a result, to cover the electrical \( 60^\circ \), 15 time-steps must be predicted at the flux and torque branch, and hence the size of torque and flux output layers remains \( 15 \times 1 \). These figures, however, can be modified as needed. The output layer for iron losses is \( 4 \times 1 \) in size. The total number of trainable network parameters is noted to be around 2.3 million. The actual number of machine designs \( N_{EM} = 44877 \) is divided into training, validation and testing. The training sample consists \( N_{EM, train} = 40390 \) \( (\sim 90\%) \), whereas validation \( N_{EM, val} = 2243 \) and test samples \( N_{EM, test} = 2244 \) comprises \( \sim 5\% \) each. As mentioned earlier in Section III-B each machine design is evaluated at 37 operating points. This means that the total number of actual training samples from number of machine design has become \( N_{D, train} = N_{EM, train} \times 37 = 1494430 \), same way, validation samples \( N_{D, val} = N_{EM, val} \times 37 = 82991 \), and test samples \( N_{D, test} = N_{EM, test} \times 37 = 83028 \).

The model training for the final configuration was carried out on Quadro M2000M GPU using standard back-propagation [18]. Tensorflow2 [19], a deep learning framework, was used to implement the training pipeline. It took around \( \sim 1 \) hour to finish the training with early stopping criteria (10 epochs over validation error) for 300 epochs. The training and validation curve is illustrated in Figure 6. To allow a fair comparison with the direct DL approach (discussed in the earlier study [1]), we used the same network architecture and model hyperparameters for DNN (Figure 9 from [1]). The model learning hyperparameters are set same for both approaches.

V. RESULTS AND ANALYSIS

In this section, we initially showcase evaluation for intermediate measures using classical machine learning and multi-branch DNN. Following that, we explain the quantitative study with empirical results of the hybrid and direct DL approach. As previously described in Section III-A the characterization of electromagnetic...
performance of PMSM is a non-linear multi-output regression problem, and thus we will evaluate the performance of meta-models using three evaluation metrics which are detailed below.

- **Mean absolute error (MAE)**: It evaluates performance measures that do not vary in different orders of magnitude and are very sensitive to small changes. The intermediate measures such as torque and flux belong to this category. MAE is defined by Equation 8 where $\hat{x}_j$ and $\tilde{x}_j$ are the true and predicted values, respectively, for the given sample. $\hat{x}_j$ can represent predicted multi-branch DNN outputs such as iron losses ($\hat{V}_{fe}$), torque ($\hat{T}$), flux ($\hat{\Psi}$) or predicted KPIs ($\hat{y}_j$) for the given machine design.

$$e_{\text{mae}}(x_j) = \frac{1}{N_{\text{test}}} \sum_{i=1}^{N_{\text{test}}} |x_j(i) - \tilde{x}_j(i)|$$  \hspace{1cm} (8)

- **Mean relative error (MRE)**: This dimensionless
Figure 5: Proposed multi-branch DNN network

Figure 6: Training curve

Figure 7: Prediction plot over test samples

metric is useful when the output measures are on different scales. It indicates how close a prediction is to the ground truth. In this study, output measures that scale differently are non-linear iron loss ($V_{fe}$) and cross-domain KPIs ($y_1, ..., y_7$).

$$
\varepsilon_{\text{me}}(x_j) = \frac{1}{N_{D_{\text{test}}}} \sum_{i=1}^{N_{D_{\text{test}}}} \frac{|x_j^{(i)} - \hat{x}_j^{(i)}|}{|x_j^{(i)}|} \times 100
$$

(9)

- Pearson correlation coefficient (PCC): Typically, this evaluation metric represents a linear relationship between variables. It indicates that the trained multi-output regression meta-model performs better when the correlation between the prediction and the ground truth is close to one $[21]$.

$$
\varepsilon_{\text{pcc}}(x_j, \hat{x}_j) = \frac{\sum_{i=1}^{N_{D_{\text{test}}}} (x_j^{(i)} - \bar{x}_j)(\hat{x}_j^{(i)} - \hat{\bar{x}}_j)}{\sqrt{\sum_{i=1}^{N_{D_{\text{test}}}} (x_j^{(i)} - \bar{x}_j)^2 \sum_{i=1}^{N_{D_{\text{test}}}} (\hat{x}_j^{(i)} - \hat{\bar{x}}_j)^2}}
$$

(10)

A. Evaluation for intermediate measures prediction

As described earlier in Section III, the prediction of intermediate measures include non-linear iron
losses ($V_{fc}$), fraction of torque ($T_{60}^y$) and flux linkages ($\Psi_{\text{coil1, }60^y}, \Psi_{\text{coil2, }60^y}, \Psi_{\text{coil3, }60^y}$) at different time-steps and each operating point for the given machine design. Accurate prediction of intermediate measures plays a vital part in hybrid approach (Figure 4) because the final calculated result is fed as input to the physics-based analytical post-processing model. The higher the accuracy, the more precise the computation of KPIs. In this paper, our main focus is on meta-modelling through the modified way of parameter-based DL approach [1] with the multi-branch DNN because it can efficiently process big data with feature automation and high prediction accuracy. However, since the problem is a multi-output regression and to allow comparison with other approaches, we first try-out other state-of-the-art multi-output regression approaches using the sci-kit-learn library [22, 23] with their default settings to train the meta-model by fitting this big size training data ($N_{\text{D,train}} = 1494430$). Here, the sci-kit-learn library contains an implementation of K-Nearest Neighbour (KNN) (with KNeighborsRegressor) [24, 25]. Support Vector Regression [26], and Gaussian Process Regression (GPR) [27], which were unable to process the data due to memory constraints. Random forest successfully trained meta-model but at the cost of accuracy and computation time. Random forest is an ensemble method, that combines multiple decision trees to train a meta-model. The final model makes predictions by taking the average of the results of the different trees (bootstrap aggregation). However, a more rigorous quantitative comparison should take hyper parameter optimization into account, but this is beyond the scope of this paper, and therefore we only used the default settings (number of estimators = 100, evaluation criterion = squared error, the minimum number of samples for each split = 2, minimum samples leaf = 1, bootstrap aggregation enabled) given in the implementation. The reason why we did not choose the classical algorithm (Random forest) is the training time and accuracy. The training time is almost $\sim 10$ higher than multi-branch DNN, and the final test accuracy is much lower compared to multi-branch DNN. The comparison between multi-branch DNN and Random forest over test samples for intermediate measures is given in Table VI and Table VII. From the results, it is evident that multi-branch DNN outperforms the prediction accuracy of Random forest. We use MRE to evaluate iron losses because its range varies largely, while torque and flux with MAE. The reason is torque ($T_{60}^y$) and flux ($\Psi_{\text{coil1, }60^y}, \Psi_{\text{coil2, }60^y}, \Psi_{\text{coil3, }60^y}$) are sensitive and do not scale big. For the testing, MAE is evaluated over the mean of predicted time steps for each operating point. Figure 7 depicts the prediction curve over each operating point for all the test samples. It is observed that multi-branch DNN predicts with high accuracy close to the ground truth for all the test samples. The evaluation time for new machine designs is $\sim 100$ ms/sample, which is much lower in comparison to the actual FE simulation. Figure 10 illustrates two operating points for different input electrical excitation conditions for one test machine design. Figure 10 displays that the multi-branch DNN has poor prediction accuracy for the operating point at zero input phase current and open circuit (no load) condition.

### B. Quantitative analysis

#### Table VII: Evaluation : hybrid and direct DL approach

| KPIs | Evaluation over test samples |
|------|-----------------------------|
|      | Hybrid approach | Direct DL approach |
| $y_1$ | $\varepsilon_{\text{MRE}}$ | $\varepsilon_{\text{MRE}}$ | $\varepsilon_{\text{PCC}}$ | $\varepsilon_{\text{PCC}}$ |
| 0.352 | 1 | 0.388 | 1 |
| 0.338 | 1 | 0.379 | 1 |
| 0.605 | 0.994 | 0.549 | 1 |
| 1.62 | 0.99 | 3.416 | 0.98 |
| 0.143 | 1 | 0.261 | 1 |
| 0.129 | 1 | 0.155 | 1 |
| 2.934 | 0.99 | 6.06 | 0.98 |

The proposed hybrid approach Figure 4 is quantitatively compared with the parameter-based direct DL approach described in the previous paper [1]. Figure 11 shows evaluation with MRE over an increasing training size from 5% ($N_{\text{EM,train},5} = N_{\text{EM,train}} \times 0.05 = 2019$) to the total training size $N_{\text{EM,train}} = 40390$. The hybrid approach consistently performs better for KPIs $y_1, y_2, y_4, y_5, y_6, y_7$, while the direct DL approach is slightly more accurate for $y_1$. Both meta-models predict KPIs for unseen machine designs ($N_{\text{EM,test}} = 2244$). Figure 12 illustrates the prediction plot over the test samples for meta-models trained on the full training set. As explained in Figure 4 the training of multi-branch DNN is independent of the system parameters ($S$) and solely relied on varying geometry, electrical excitation, and material parameters, whereas in the direct DL approach, model training with output KPIs implicitly involve fixed value for system parameters ($S$). This makes the hybrid approach more flexible compared to the direct DL approach which becomes deterministic because of supervised training of DNN directly onto the final KPIs. The post-processing of FE output takes very little time and is performed with a physics-based
customized in-house post-processing tool, and hence hybrid approach opens up new avenues for electrical machine analysis. With the high calculation capacity of the post-processing tool, one can evaluate different performance curves and efficiency maps which yields a thorough analysis of the electrical machine from the electromagnetic performance point of view. Demonstrative example for one test sample is shown in Figure 9 and Figure 8. Figure 9 displays different performance curves, e.g., maximum torque curve, open circuit, and short circuit voltage characteristic, and maximum shaft power at different rotor speeds. In the post-processing tool, the efficiency of the electrical machine for the desired operating point (torque and speed) is calculated by using detailed finite-element mapping of losses, torque, and flux-linkage as a function of the d-axis and q-axis currents and speed. It is simply a ratio of average input electrical power to obtained output mechanical power of the machine during one cycle in the motor mode operation and vice versa in the generator mode. The detailed interpretation of the efficiency map is given in [29]. Figure 8 shows the efficiency map for the given test design. The Figure 8a presents an efficiency map for the actual FE simulation, Figure 8b is calculated from the intermediate measures predicted with multi-branch DNN, while Figure 8c gives details about the deviation between the two. The difference is around zero and a little bit high somewhere near the low torque region (20%). The possible reason could be that multi-branch DNN does not predict well for the open circuit operation mode at zero input current, which is also reflected in Figure 10b for the same associated sample. In the open circuit mode of operation, the associated torque range for the electrical machine is significantly low (≈ 1e^{-1} – 1e^{-3} Nm). As a result, in that condition, if the multi-branch DNN predicts values with the deviation of ≈ 1e^{-2} – 1e^{-4} Nm results in a very high relative error. However, relatively poor prediction in this region does not make much impact on calculating the overall efficiency of the electrical machine at other operating points (especially high-efficiency operating point regions as shown in Figure 8c).

There are also some disadvantages of the proposed hybrid approach as compared to the direct DL approach. The training time of the multi-branch DNN is roughly about ~ 1 hour, which is about 6× higher than the training time (~ 10 minutes) of the DNN defined for direct KPIs prediction [1]. The possible explanation for that the multi-branch DNN in the hybrid approach has
to deal with a high number of samples for model training (37× compared to the direct DNN in this study), and also the number of model parameters is around ~ 2.3 million. Therefore, the hybrid approach requires higher computational power compared to DNN for the direct KPIs prediction. The time to estimate KPIs for the hybrid approach using a post-processing tool for analyzing new electrical machine design is ~ 3−5 minutes, while the trained DNN for the direct DL method takes ~ 1ms/sample. However, the computational time is much lower in comparison with magneto-static FE simulation, which takes around ~ 3 − 5 hours on a single-core CPU. In [Section III] certain conditions are described under which the magnetic state symmetry of the machines is exploited to reduce the computational cost, while this is not the case for another approach. In the hybrid method, if the magnetic state symmetry of the electrical machine is not maintained, then possibly different DNNs for each output intermediate measures prediction have to be trained or need to search for a network architecture that can handle a very large number of outputs (in this study we have handled 72 outputs). It may require very high computational power and may limit the real-world application of the proposed approach.

VI. CONCLUSION

In this paper, we have presented a data-driven DL-based hybrid method for the performance analysis of the electrical machine. The dataset used for the research is generated from a real-world industry design workflow. The problem is formulated as a non-linear multi-output
regression, and the DL meta-model is trained using supervised learning. The intermediate measures (iron losses, flux, and torque) represent the primary characteristics of an electromagnetic system for the electrical machine. The multi-branch DNN is introduced to predict the intermediate measures close to the ground truth values. It is designed to handle a large number of FE outputs and a high-dimensional parameter space. The proposed hybrid approach is better than the previously defined parameter-based direct DL approach in terms of KPIs estimation and flexibility. This gain can be explained by two facts. Firstly, learning a few time-steps of intermediate measures is expected to be simpler than multiple (possibly independent) cross-domain KPIs and secondly, the post-processing tool exploits the laws of physics and thus ensuring that the KPIs fulfill the right constraints. Also on the application side, this approach has advantages. It makes the calculation of KPIs independent from the system parameters by predicting intermediate measures during the optimization and allows the analysis of electrical machines with more complex performance measures such as the efficiency map, different characteristic curves (maximum torque, open-circuit voltage, short circuit current, etc.) using a powerful physics-based post-processing tool. We have demonstrated that the trained multi-branch DNN meta-model evaluates new unseen designs much faster and with less computational effort than the magneto-static FE simulation. The predictability of the meta-model also depends on its hyperparameters. Therefore, we performed HPO with an in-house optimization tool to obtain the best possible model configuration. In future work, the proposed hybrid approach can be applied to many query scenarios, such as multi-objective optimization or uncertainty quantification. An interesting research direction for the future could be to find a strategy when the magnetic state symmetry is not present in the design of electrical machines.

REFERENCES

[1] V. Parekh, D. Flore, and S. Schöps, “Deep learning-based prediction of key performance indicators for electrical machines,” IEEE Access, vol. 9, pp. 21 786–21 797, 2021. DOI: 10.1109/ACCESS.2021.3053856.

[2] S. Tang, S. Yuan, and Y. Zhu, “Deep learning-based intelligent fault diagnosis methods toward rotating machinery,” IEEE Access, vol. 8, pp. 9335–9346, 2020. DOI: 10.1109/ACCESS.2019.2963092.

[3] Y.-B. Yan, J.-N. Liang, T.-F. Sun, J.-P. Geng, Gang-Xie, and D.-J. Pan, “Torque estimation and control of pmsm based on deep learning,” in 2019 22nd International Conference on Electrical Machines and Systems (ICEMS), 2019, pp. 1–6. DOI: 10.1109/ICEMS.2019.8921886.

[4] H. Kurtović and I. Hahn, “Neural network meta-modeling and optimization of flux switching machines,” in 2019 IEEE International Electric Machines Drives Conference (IEMDC), 2019, pp. 629–636. DOI: 10.1109/IEMDC.2019.8785344.

[5] S. Doi, H. Sasaki, and H. Igarashi, “Multi-objective topology optimization of rotating machines using deep learning,” IEEE Transactions on Magnetics, vol. 55, no. 6, pp. 1–5, 2019. DOI: 10.1109/TMAG.2019.2899934.

[6] A. Khan, M. H. Mohammad, V. Ghobarian, and D. Lowther, “Efficiency map prediction of motor drives using deep learning,” IEEE Transactions on Magnetics, vol. 56, no. 3, pp. 1–4, 2020. DOI: 10.1109/TMAG.2019.2957162.

[7] H. Sasaki, Y. Hidaka, and H. Igarashi, “Explainable deep neural network for design of electric motors,” IEEE Transactions on Magnetics, vol. 57, no. 6, pp. 1–4, 2021. DOI: 10.1109/TMAG.2021.3063141.
[8] A. Khan and D. A. Lowther, “Machine learning applied to the design and analysis of low frequency electromagnetic devices,” in 2020 21st International Symposium on Electrical Apparatus Technologies (SIELA), 2020, pp. 1–4. DOI: 10.1109/SIELA49118.2020.9167158

[9] Y. Li, G. Lei, G. Bramerforder, S. Peng, X. Sun, and J. Zhu, “Machine learning for design optimization of electromagnetic devices: Recent developments and future directions,” Applied Sciences, vol. 11, no. 4, 2021, ISSN: 2076-3417. DOI: 10.3390/app11041627

[10] N. Gabdullin, S. Madanzadeh, and A. Vilkin, “Towards end-to-end deep learning performance analysis of electric motors,” Actuators, vol. 10, no. 2. 10.3390/act10020028, 2021.

[11] S. Barmada, N. Fontana, A. Formisano, D. Thomopulos, and M. Tucci, “A deep learning surrogate model for topology optimization,” IEEE Transactions on Magnetics, vol. 57, no. 6, pp. 1–4, 2021. DOI: 10.1109/TMAG.2021.3063470

[12] S. Barmada, N. Fontana, L. Sani, D. Thomopulos, and M. Tucci, “Deep learning and reduced models for fast optimization in electromagnetics,” IEEE Transactions on Magnetics, vol. 56, no. 3, pp. 1–4, 2020. DOI: 10.1109/TMAG.2019.2957197

[13] J. Asanuma, S. Doi, and H. Igarashi, “Transfer learning through deep learning: Application to topology optimization of electric motor,” IEEE Transactions on Magnetics, vol. 56, no. 3, pp. 1–4, 2020. DOI: 10.1109/TMAG.2019.2956849

[14] M. Tucci, S. Barmada, A. Formisano, and D. Thomopulos, “A regularized procedure to generate a deep learning model for topology optimization of electromagnetic devices,” Electronics, vol. 10, no. 18, 2021, ISSN: 2079-9292. DOI: 10.3390/electronics10182185

[15] M. Reinein, T. Hubert, A. Hoffmann, and A. Kremser, “Optimization of analytical iron loss approaches for electrical machines,” in 2013 3rd International Electric Drives Production Conference (EDPC), 2013, pp. 1–7. DOI: 10.1109/EDPC.2013.6689759

[16] H. Jordan, “Die ferromagnetischen konstanten für schwache wechselfelder,” Elektr. Nach. Techn., vol. 1, no. 8, 1924.

[17] L. Li, K. Jamieson, A. Rostamizadeh, E. Gornina, J. Ben-tzur, M. Hardt, B. Recht, and A. Talwalkar, “A system for massively parallel hyperparameter tuning,” in Proceedings of Machine Learning and Systems, I. Dhillon, D. Papailiopoulos, and V. Sze, Eds., vol. 2, 2020, pp. 230–246. [Online]. Available: https://proceedings.mlr.press/paper/2020/f4b9ec30ad9f68189b29639786cb62ef-Paper.pdf

[18] D. E. Rumelhart, G. E. Hinton, and R. J. Williams, “Learning representations by back-propagating errors,” nature, vol. 323, no. 6088, pp. 533–536, 1986. DOI: https://doi.org/10.1038/323533a0

[19] M. Abadi, A. Agarwal, P. Barham, E. Brevo, Z. Chen, C. Citro, G. S. Corrado, A. Davis, J. Dean, M. Devin, S. Ghemawat, I. Goodfellow, A. Harp, G. Irving, M. Isard, Y. Jia, R. Jozefowicz, L. Kaiser, M. Kudlur, J. Levenberg, D. Mané, R. Monga, S. Moore, D. Murray, C. Olah, M. Schuster, J. Shlens, B. Steiner, I. Sutskever, K. Talwar, P. Tucker, V. Vanhoucke, V. Vasudevan, F. Viégas, O. Vinyals, P. Warden, M. Wattenberg, M. Wicke, Y. Yu, and X. Zheng, TensorFlow: Large-scale machine learning on heterogeneous systems, 2015. [Online]. Available: https://www.tensorflow.org/

[20] E. M. Dogo, O. J. Afolabi, N. I. Nwulu, B. Twala, and C. O. Aigbavboa, “A comparative analysis of gradient descent-based optimization algorithms on convolutional neural networks,” in 2018 International Conference on Computational Techniques, Electronics and Mechanical Systems (CTEMS), 2018, pp. 92–99. DOI: 10.1109/CTEMS.2018.8769211

[21] “Pearson’s correlation coefficient,” in Encyclopedia of Public Health, W. Kirch, Ed. Dordrecht: Springer Netherlands, 2008, pp. 1090–1091, ISBN: 978-1-4020-5614-7. DOI: 10.1007/978-1-4020-5614-7_2569

[22] F. Pedregosa, G. Varoquaux, A. Gramfort, V. Michel, B. Thirion, O. Grisel, M. Blondel, P. Prettenhofer, R. Weiss, V. Dubourg, J. Vanderplas, A. Passos, D. Cournapeau, M. Brucher, M. Perrot, and É. Duchesnay, “Scikit-learn: Machine learning in python,” Journal of Machine Learning Research, vol. 12, no. 85, pp. 2825–2830, 2011. [Online]. Available: http://jmlr.org/papers/v12/pedregosa11a.html

[23] L. Büitinck, G. Louppe, M. Blondel, F. Pedregosa, A. Mueller, O. Grisel, V. Niculae, P. Prettenhofer, A. Gramfort, J. Grobler, R. Layton, J. Vanderplas, A. Joly, B. Holt, and G. Varoquaux, “API design for machine learning software: experiences from the scikit-learn project,” in European Conference on Machine Learning and Principles and Practices of Knowledge Discovery in Databases, 2013. [Online]. Available: https://hal.inria.fr/hal-00856511/file/paper.pdf

[24] S. M. Omohundro, “Five balltree construction algorithms,” Tech. Rep., 1989.

[25] J. L. Bentley, “Multidimensional binary search trees used for associative searching,” Commun. ACM, vol. 18, no. 9, pp. 509–517, Sep. 1975, ISSN: 0001-0782. DOI: 10.1145/361002.361007
[26] C.-C. Chang and C.-J. Lin, “Libsvm: A library for support vector machines,” *ACM Trans. Intell. Syst. Technol.*, vol. 2, no. 3, May 2011, ISSN: 2157-6904. DOI: [10.1145/1961189.1961199](https://doi.org/10.1145/1961189.1961199)

[27] C.E.Rasmussen and C. Williams, “Gaussian processes for machine learning,” in *Adaptive Computation and Machine Learning*, the MIT Press, 2006, pp. 7–30, ISBN: 026218253X. [Online]. Available: [http://www.gaussianprocess.org/gpml/](http://www.gaussianprocess.org/gpml/)

[28] L. Breiman, “Random forests,” *Machine learning*, vol. 45, no. 1, pp. 5–32, 2001.

[29] A. Mahmoudi, W. L. Soong, G. Pellegrino, and E. Armando, “Efficiency maps of electrical machines,” in *2015 IEEE Energy Conversion Congress and Exposition (ECCE)*, 2015, pp. 2791–2799. DOI: [10.1109/ECCE.2015.7310051](https://doi.org/10.1109/ECCE.2015.7310051)