Inductor Magnetic Energy Estimation Based on Multilayer Perceptron Regression Model

UNIVERSITÉ LAVAL

Faculté des sciences et de génie

Group 23: David Cajander, Ayoub Echchahed, Majid Heidary, Nassim El Massaudi, Tak Wa Ng

GIF 7005 / GIF 7015 Introduction to Machine Learning

Inductor type C2B2mEn

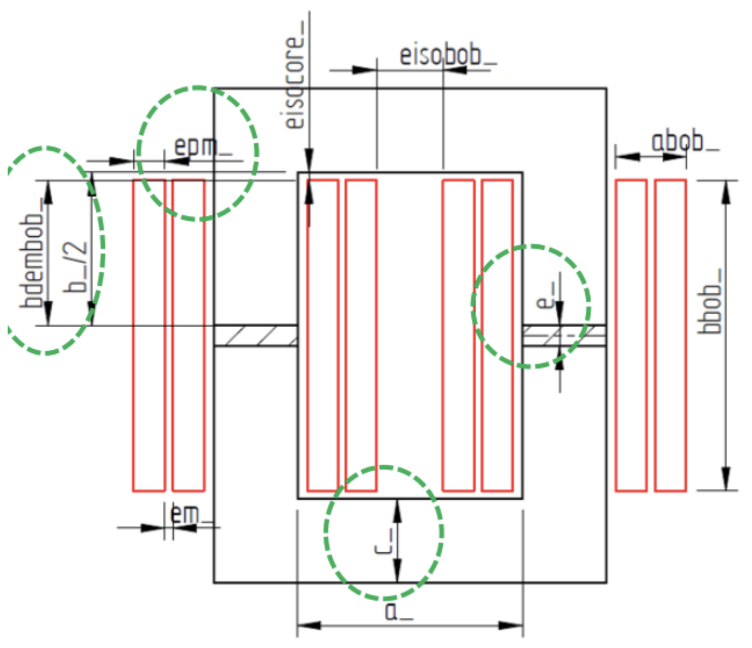
Introduction : Context and Objective

Context : This project aims to contribute to the field of the dimensioning and modeling of electromagnetic devices (the type of DC power supply filter inductors 0-13000A of 2.47MW) by replacing an electromagnetic model based on finite element analysis (FEA, a highly time-consuming numerical partial differential equation (PDE) method) with a data-driven supervised model to forecast the inductor magnetic energy based on the inductor's physical dimensions and the magnetic core's induction to reduce the computational cost significantly with acceptable precision of inductor magnetic energy estimation.

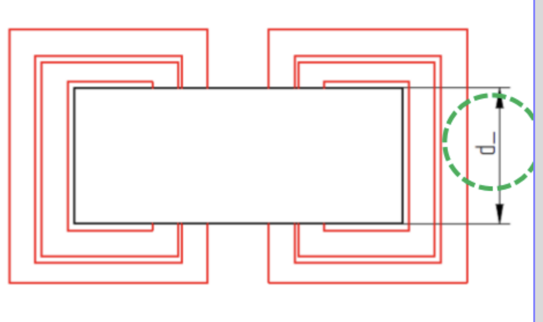
Objective : To explore and compare different supervised regression models : multilayer perceptron (MLP) regression, linear/polynomial regression, and decision tree regression, as the replacement for the PDE model based on the FEA.

Dataset : 10^6 samples, 7 features and 1 target

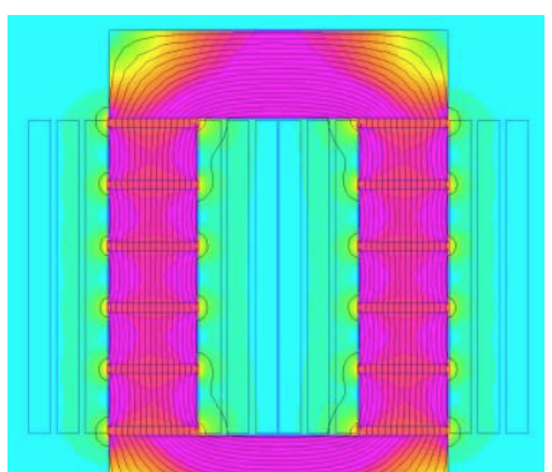
- 5 physical dimensions features : the topology of the inductor (*e_, c_, d_, epm_, bdembob_*, unit : m)
- 1 feature : the volume of the inductor (unit : m^3)
- 1 feature : the maximum induction in the magnetic core (Unit : Tesla)
- 1 target : the magnetic energy contained in the inductor (unit : Joule)



Inductor side view

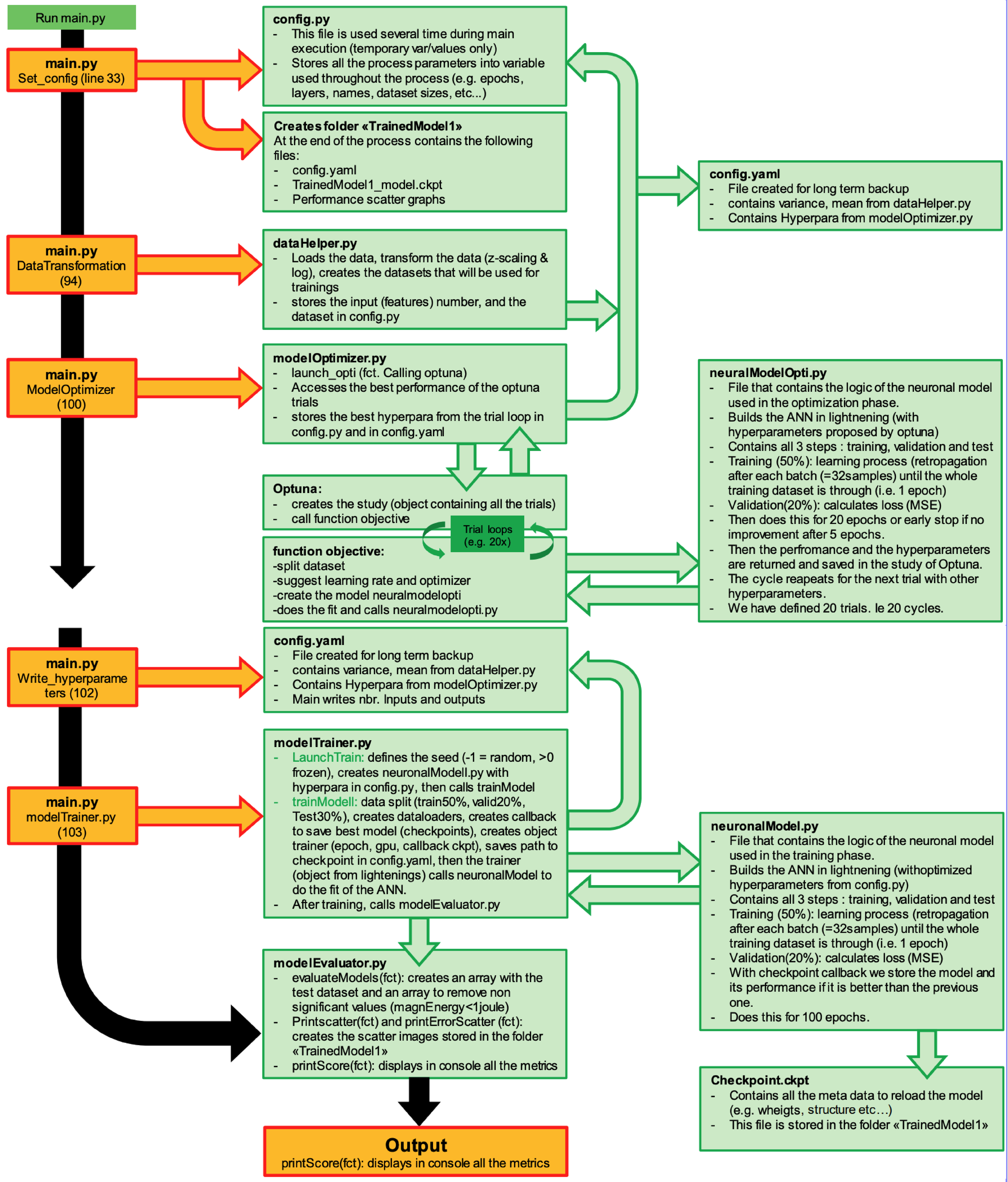


Inductor view from above



Flux density in magnetic core

Methodolgy and Demonstration of Main Model : MLP Regression

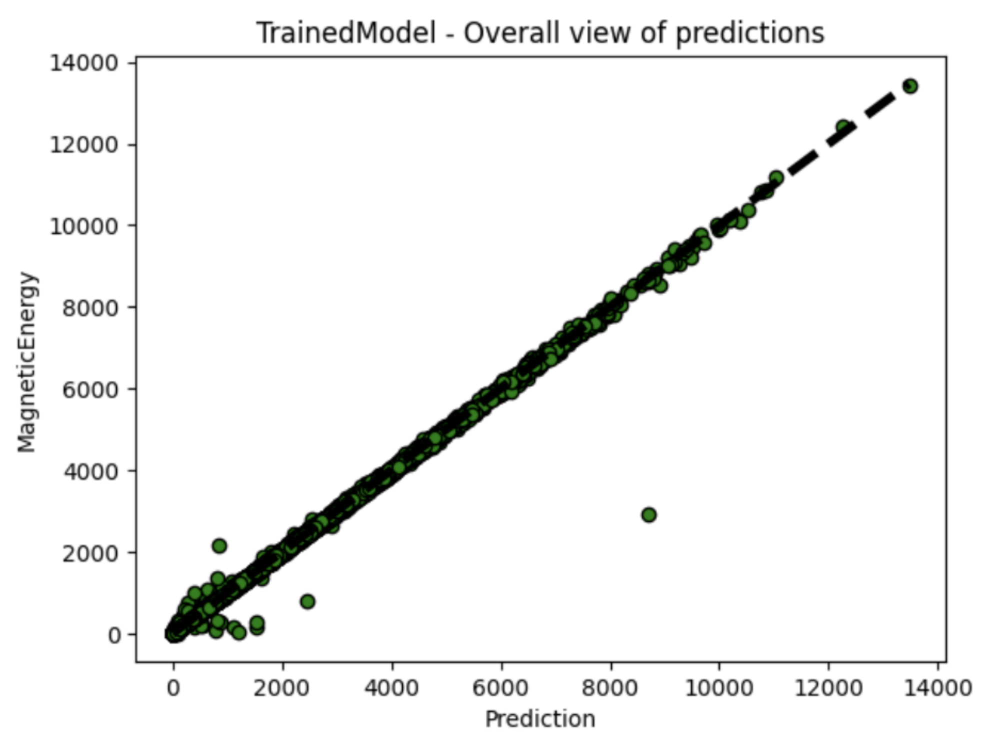


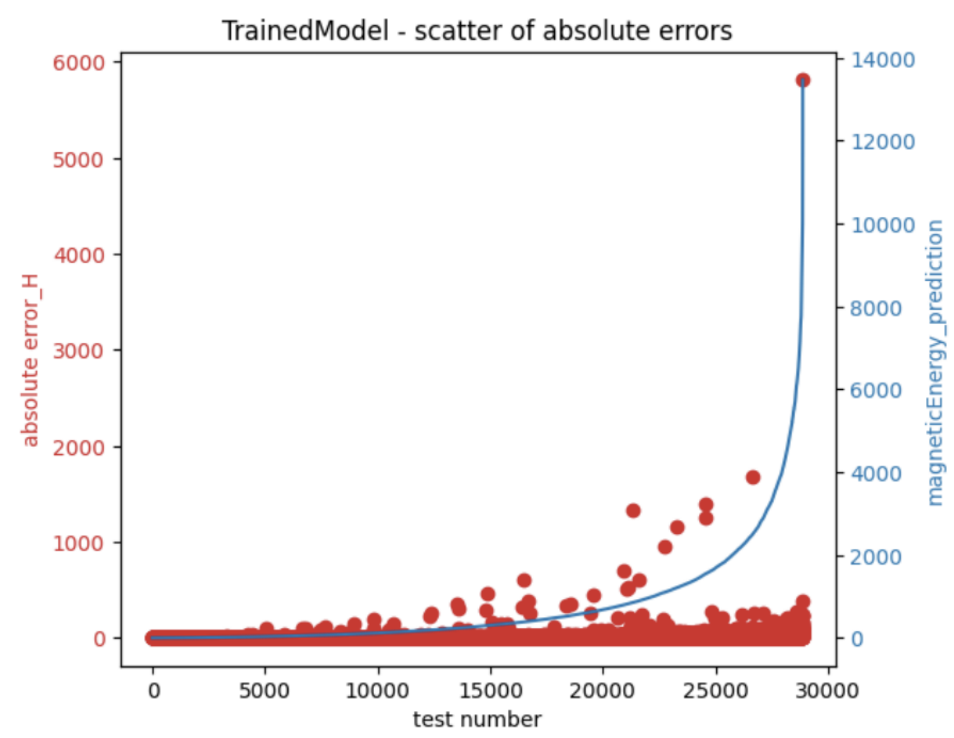
Results : Model Comparison and MLP Prediction with Score 1.956%

```
n_inputs: 7
n_output: 1
n_layers: 0
learning_rate: 0.000794986
n_hidden: [476]
optimizer: Optimizer.Adam
checkpoint_path: TrainedModel_model.ckpt
Optimization Epochs: 20
Optimization Trials: 100
Training Epochs: 100
Training dataset split: 0.5
Validation dataset split: 0.2
Test dataset split: 0.3
Pearsons coefs r2: 0.9986
Forecast relative error: 1.956%
```

Best hyperparameter configuration from Optuna after **100 trials (best hyperparam.)** :
lr=0.0007949
hidden layers = 0 (ie 1 layer)
Neurones per layer = 476
Training epochs: **100**

```
Derivation coeficient r2 (Pearson):0.9986
Minimum relative error : 1.629e-04%
Maximum relative error : 312.004%
Mean relative error : 1.956%
Median relative error : 1.28%
```





Models / Metrics	Accuracy (R^2)	Accuracy without Data Preprocessing (R^2)
Multilayer Perceptron	0.9986	0
Linear Regression	0.7918	0.5974
Polynomial Regression	0.9567	0.9213
Decision Tree Regression	0.9873	0.9723

Discussion and Conclusion : MLP as the Best Candidate

We obtain fast convergence for all model with high accuracies (only if the data preprocessing is applied). After comparing and analyzing the results like testing with different dataset sizes for understanding which models perform better in a sparse context and testing the extrapolation regime error for each model, our model of choice is the MLP.

Decision Tree vs MLP : Even if the decision tree looks like a competitive candidate, its result is obtained in a simplified problem with only 7 features and 1 target. With more features as the input, we believe that the MLP can perform better than the decision tree, as the MLP should be more suitable for larger dataset with complex non-linear mapping in high dimensions.

Important and interesting observations :

1. **Data preprocessing :** Without the z-scaling on features (removing unit and standardizing the features) and the log transformation on the target (minimizing the skewness), the training showed poor performance, especially for MLP.
2. **Hyperparameters trials of MLP :** The processing time is long but that is the key to obtaining a good-performing ANN model. The best hyperparameter set has been provided after 100 trials using *Optuna*. Without proper range setting and/or enough trial, *Optuna* might not be able to provide good hyperparameters.
3. **The number of training epochs in MLP :** It has been set to 100 to reach a score below 2% (10 epochs for 20% and 20 epochs for 5%).