

**Technical University of Applied Sciences Würzburg-Schweinfurt
(THWS)**

Faculty of Computer Science and Business Information Systems

Master Thesis

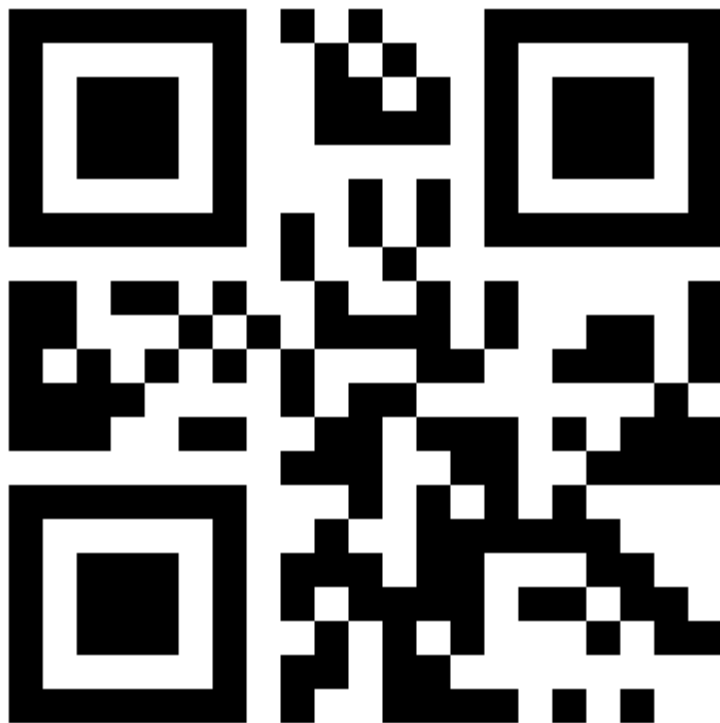
Electric Motor Modelling via Graph Neural Networks

Submitted to the Technical University of Applied Sciences
Würzburg-Schweinfurt in the Faculty of Computer Science and Business
Information Systems to complete a course of studies in Master of Artificial
Intelligence

Lilly Abraham
K64889

To be Submitted on: 11.12.2024

Initial examiner: Prof. Dr. Magda Gregorova
Secondary examiner: Prof. Gracia Herranz Mercedes



Abstract

The thesis explores an approach to predict KPIs of topology invariant IPSM Electric Motors by transforming its geometric, physical and simulation parameters into a graph representation.

The KPIs to be predicted are plots on Efficiency grid(3D) and Torque curve(2D).

We aim to first parameterize the EM design such that it is feasible to convert into a graph representation.

Next, we would create a Graph with relevant attributes and design a GNN with the graph as input and the plots in the format of vectors as target values.

Additionally we may also need to customize the loss function in a way that would smoothen out the plot curves of the prediction values.

Then, we would evaluate the predictions with the test target values by experimenting with various hyperparameter tuning settings and as a baseline with an MLP model of the parameters in tabular form.

Finally we will enable the KPI's plot visualisation in a manner presentable to the client Valeo(Automaker Company).

Not necessary remove i suppose.... The aim of the Master Thesis is to train a neural network to learn the parameters of Electric Motors and thus be able to predict its KPIs. The KPIs are 2D and 3D plots on Torque(Mgrenz) curve(Mgrenz) and Efficiency grid(ETA). Other KPIs can be calculated from these two KPIs. For instance the Vibration Costs are inversely proportional to the Efficiency values predicted.

Acknowledgement

I would like to thank my supervisor Prof. Dr. Magda Gregorova for her guidance and support throughout the course of this thesis and Valeo for providing the data. Special thanks to Mr Daniel and Leo for sharing valuable insights of the data from an electromechanical standpoint.

Contents

Abstract	2
Acknowledgement	3
Contents	4
Abbreviations	5
Introduction	6
Literature Review	8
Dataset	9
Scaling	12
Dataset splitting	13
Modelling	14
Multi Linear Perceptron Model	14
HeterogeneousGraph Neural Network Model	14
Loss Function	14
Loss Regularization for 2D KPI(Torque curve)	14
Loss Customization for 3D KPI(Efficiency Grid)	15
Evaluation Metrics	15
Experiments and Results	16
List of Figures	22
List of Tables	23
Appendix	24
Bibliography	25
Declaration on oath	26
Consent to Plagiarism Check	27

Abbreviations

GNN : Graph Neural Network

MLP : Multi Linear Perceptron

GNN : Graph Neural Network

KPI : Key Performance Indicator

EM : Electric Motor

FEM : Finite Element Method

CNN : Convolution Neural Network

2D : 2 Dimension

3D : 3 Dimension

Introduction

In the design of electric motors, vast amounts of data are generated to determine which design of an EM fits best to KPIs.

KPIs of an Electric Motor are essential to judge the performance of the motor before it is manufactured. Traditionally these KPIs are inferred from a description of an EM design via a FEM approximating the solutions of the Maxwell's equations. This process, though well established in the EM design, is very time consuming and does not allow for high-throughput engine design optimization.

The actual engine data of Valeo is used here as the dataset comprising of multiple variant designs of the Double-V topology.

The 3 motor topologies manufactured by Valeo are as below:

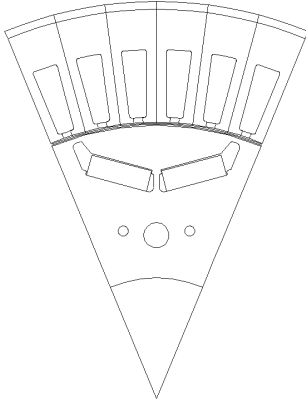


Figure 1: V1 Magnet
(Source:Valeo)

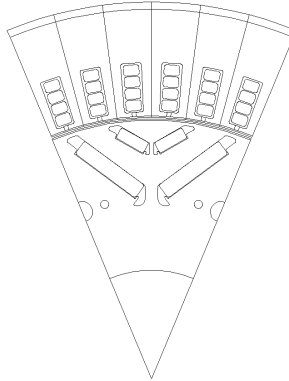


Figure 2: V2 Magnet
(Source:Valeo)

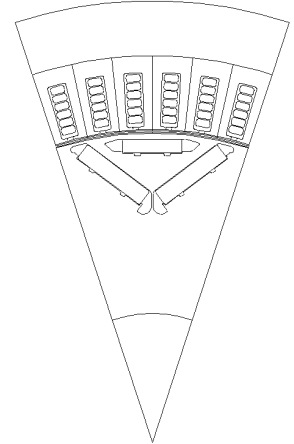


Figure 3: Nabla Magnet
(Source:Valeo)

This master thesis explores a way to do surrogate modelling of the current process as is highlighted in Figure 4 by making use of GNN/MLP for the modelling of electrical engine designs described parameterically.

The thesis is structured to follow sections namely Literature Review, Dataset, Modelling, Experiments and Results, Conclusion, Appendix and Bibliography.

The Literature Review section will cover the related works that has been done in this domain. In the Dataset section a detailed insight to how our data is structured is given. In the Modelling section, the methodologies used to tackle the problem will be further elaborated on. The Experiments and Results chapter gives an outlook on the outcomes of our work in addition to other findings we unearth. Conclusion chapter summarizes the thesis briefly and would also give a small glimpse into areas of improvement. Finally the Bibliography section lists out the articles cited for this thesis.

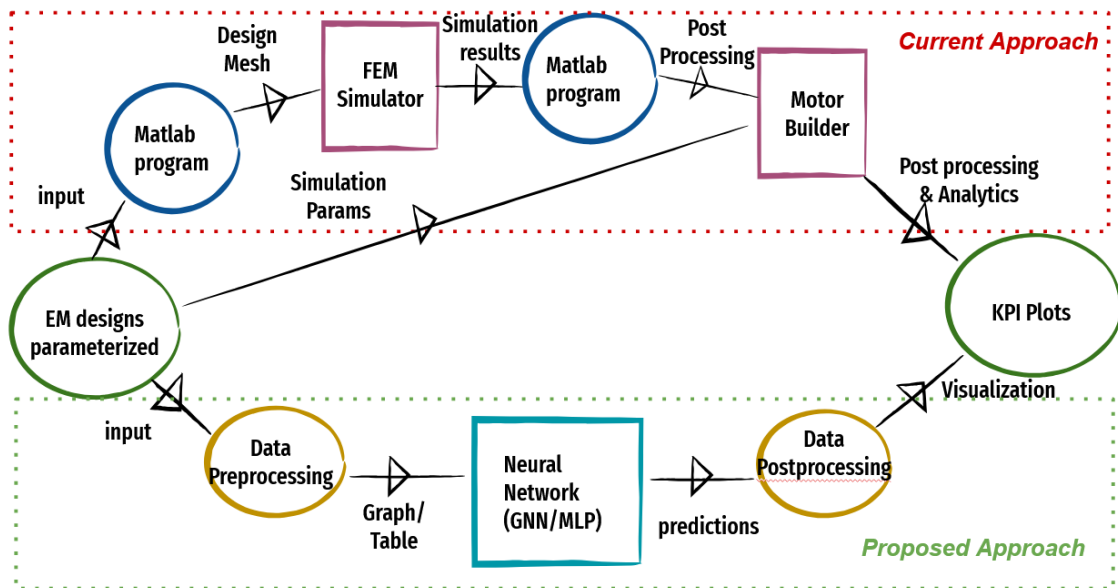


Figure 4: EM Design Flowchart

Literature Review

There has been extensive research in modeling the Electric Motor with CNN based on the images of the motor cross-section. However our approach is progressive in the sense that once the KPIs are predicted we would like to be able to generate the inputs and reproducing images is not known to apt given the infamous known fact that AI generated images are faulty. Instead by generating the parameters of the motor we can be rest assured of more precise results. Hence the need to focus on the inputs as they are with their parametric description. Existing literature also covers works on modelling this work as tabular data using MLPs. Although this is fairly good forseeing the impact of generating the inverse process yet MLPs cannot necessarily learn all the intricacies within motor components. Hence the need to better represent the data typically in the form of graphs and model Graph Neural Networks to achieve the desired results. There has been close to no work of GNNs in this domain. However we see progress of GNNs in molecular chemistry and social networks usecases from which we draw inspiration.

Dataset

Valeo an automotive company has supplied the dataset consisting of close to 1500 Double V Electric Motor parameters. There are close to 196 parameters which comprises of the geometric, physical and simulation properties of the motor.

The geometry of a whole Double V motor is as below

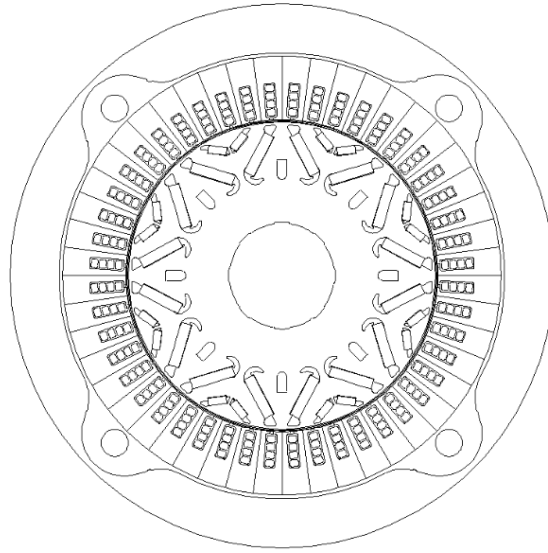


Figure 5: Complete EM Geometry(Source:Valeo)

Below is the geometry of 1/8 cross-section of the same motor.

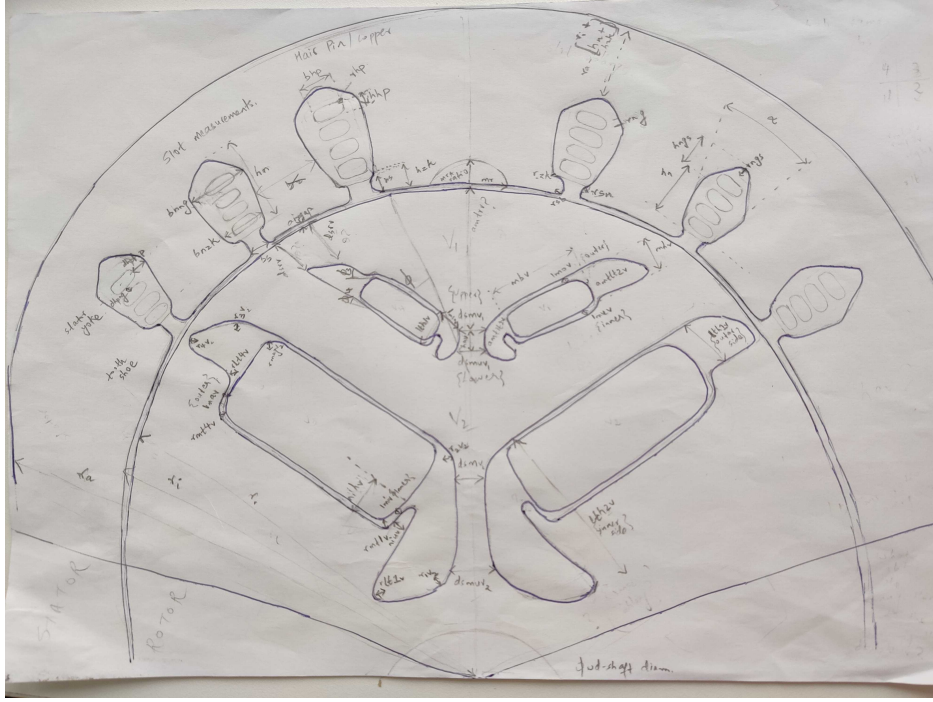


Figure 6: 1/8 Motor Crossection

For modelling the GNN, we represent the data in the form of a heterogeneous graph with different node and edge types.

Node types

1. General

- General parameters:

$$r = \{r_i\} \quad \forall i \in \{a, r, o\}$$

where:

- r_a : Outer Radius of the Stator
- r_r : Outer Radius of the Rotor
- r_o : Center of the EM

2. Stator

- Slot windings:

$$sw = \{s_i w_j\} \quad \forall i \in \{1, \dots, QSim\}, \quad \forall j \in \{1, \dots, N\}$$

- Slots:

$$s = \{s_i\} \quad \forall i \in \{1, \dots, QSim\}$$

where

- $Qsim$: Count of slots in the Stator
- N : Count of copper windings per slot

3. Rotor

- Magnet Flux Barriers:

$$v = \{v_{ij}\} \quad \forall i \in \{1, \dots, T\}, \quad \forall j \in \{1, \dots, V\}$$

- Magnets:

$$vm = \{v_i m_j\} \quad \forall i \in \{1, \dots, T\}, \quad \forall j \in \{1, \dots, V\}$$

where

- T : Topology type of the EM
- V : Type of Magnet

As Valeo only manufactures Double V magnets we consider it to be 2

Edge types

1. Angle

Relevant Paths

$$vm - -vm = \{v_{i_1} m_{j_1} - v_{i_2} m_{j_2}\} \forall i_1, i_2 \in \{1, \dots, T\}, \quad \forall j_1, j_2 \in \{1, \dots, V\} \mid i_1 = i_2, \quad j_1 \neq j_2$$

$$\text{angle} = vm - vm$$

2. Distance

Relevant Paths

$$vi - -vi = \{v_{i_{j_1}} - v_{i_{j_2}}\}, \forall i \in \{1, \dots, T\}, \forall j_1, j_2 \in \{1, \dots, V\} \mid j_1 \neq j_2$$

$$vi - -vj = \{v_{i_{j_1}} - v_{i_{j_2}}\}, \forall i_1, i_2 \in \{1, \dots, T\}, \forall j \in \{1, \dots, V\} \mid i_1 \neq i_2$$

$$v - -vm = \{v_{ij} - v_i m_j\} \forall i \in \{1, \dots, T\}, \quad \forall j \in \{1, \dots, V\}$$

$$v - -rr = \{v_{ij} - r_r\}, \forall i, j \in \{1, \dots, T\}$$

$$o - -r = \{(o - r_r), (o - r_a)\}$$

$$rr - -s = \{r_r - s_i\}, \forall i \in \{1, \dots, QSim\}$$

$$s - -sw = \{s_i - s_i w_j\}, \forall i \in \{1, \dots, QSim\}, \forall j \in \{1, \dots, N\}$$

$$s - -ra = \{s_i - r_a\}, \forall i \in \{1, \dots, QSim\}$$

$$sw - -sw = \{s_i w_{j_1} - s_i w_{j_2}\}, \forall i \in \{1, \dots, QSim\}, \forall j \in \{1, \dots, N\} \mid (j_1 == j_2 - 1)$$

$$\text{distance} = vi - vi + vi - vj + v - vm + v - rr + o - r + rr - s + s - sw + s - ra + sw - sw$$

Node Features

$$1. \mathbf{v} = \{\text{lmsov}, \text{lth1v}, \text{lth2v}, \text{r1v}, \text{r11v}, \text{r2v}, \text{r3v}, \text{r4v}, \text{rmt1v}, \text{rmt4v}, \text{rlt1v}, \text{rlt4v}, \text{hav}\}$$

$$2. \mathbf{vm} = \{\text{mbv}, \text{mhv}, \text{rmagv}\}$$

$$3. \mathbf{r} = \{r\}$$

$$4. \mathbf{s} = \{\text{b_nng}, \text{b_nzk}, \text{b_s}, \text{h_n}, \text{h_s}, \text{r_sn}, \text{r_zk}, \text{r_ng}, \text{h_zk}\}$$

$$5. \mathbf{sw} = \{\text{bhp}, \text{hhp}, \text{rhp}\}$$

Path Features

$$1. \mathbf{vm - vm} = \{\text{deg_phi}\}$$

$$2. \mathbf{vi - vi} = \{\text{dsm}, \text{dsmu}\}$$

$$3. \mathbf{vi - vj} = \{\text{amtrvj - amtrvi}\}$$

4. $\mathbf{v}-\mathbf{vm} = \{\text{lmav}, \text{lmiv}, \text{lmov}, \text{lmuv}\}$
5. $\mathbf{v}-\mathbf{r} = \{\text{amtrv}, \text{dsrv}\}$
6. $\mathbf{o}-\mathbf{r} = \{\mathbf{r}\}$
7. $\mathbf{rr}-\mathbf{s} = \{\text{airgap}\}$
8. $\mathbf{s}-\mathbf{sw} = \{\text{dhphp}\}$
9. $\mathbf{sw}-\mathbf{sw} = \{\text{dhpng}\}$
10. $\mathbf{s}-\mathbf{ra} = \{\mathbf{r}_a - (\mathbf{r}_i + \mathbf{h}_n + \mathbf{h}_{zk})\}$

The heterogeneous graph that was constructed earlier is as below:

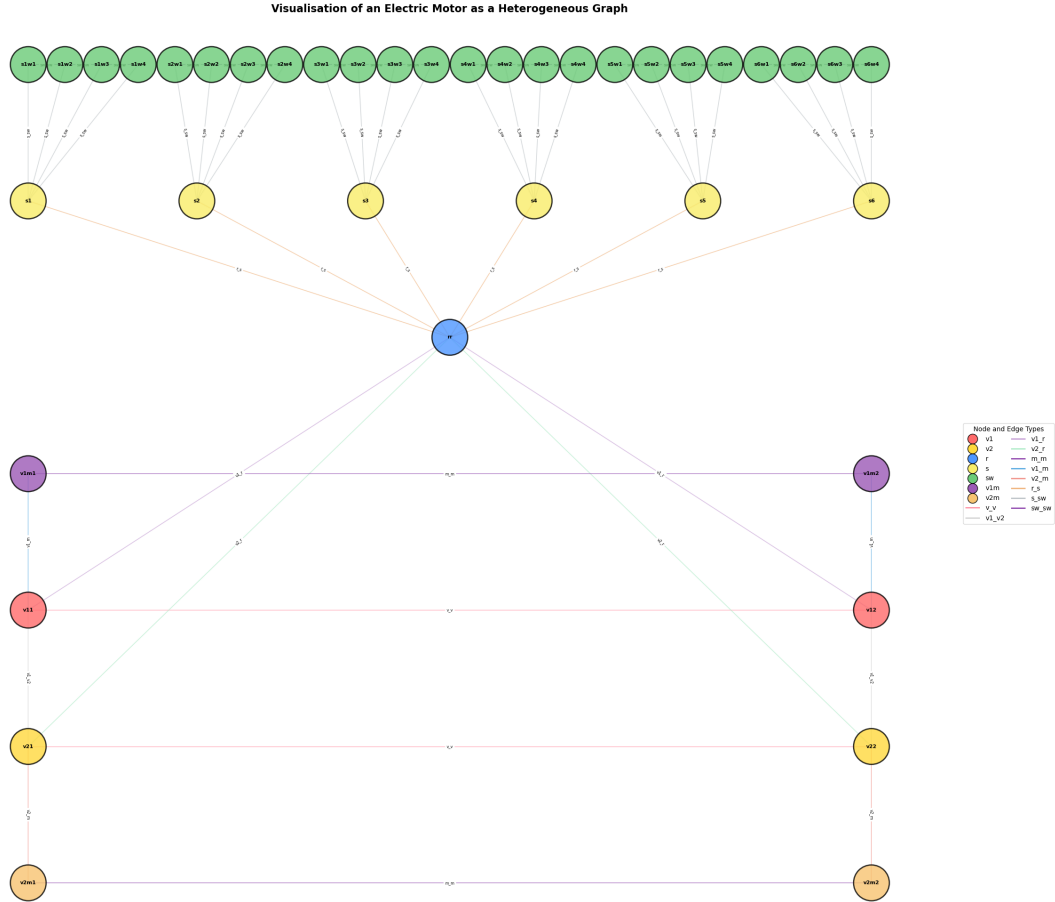


Figure 7: HetGraph

Scaling

To scale the input features, we have used the standard scaler from the sklearn library to scale the values to be shifted such that it is of 0 mean and unit standard deviation. The target values however are scaled with the MinMax Scaler from sklearn to be between 0 and 1. The scalers are applied to the train validation datasets and the same scaler is used to transform the test dataset to maintain uniformity on the predictions generated.

Dataset splitting

We have also split the dataset to have about 50 samples for test and the remaining is used for 5 fold cross validation with 80:20 split for training and validation. Within 5 folds, we expect to cover most grounds on training and have good monitoring on the model's performance for each fold. The best performing model is then chosen based on the evaluation metrics we have in place and are used for inference later on.

Modelling

Since we aim to predict continuous vector values, we model this task into a regression problem

As a baseline, we first train a MLP on the tabular representation of the data and work on it further to do the same with a heterogeneous GNN.

Multi Linear Perceptron Model

For the Multi Linear Perceptron (MLP) model, we use a single model with input features corresponding to all the features in the tabular topology invariant representation of the data and feed forward layers. The model architecture is designed to predict both the 2D and 3D KPIs by have 2 separate output layers for each of the KPIs. Since the 2D KPI's targets are relatively learnable then that of the 3D KPI's targets we have experimented with fewer feed forward layers in the former than in the latter. RELU layers were also added in between to serve as the activation function and produce non-linearities in the model.

HeterogeneousGraph Neural Network Model

We find the heterogeneous graph to be most apt for our use case with its different node and edge types. As it preserves both the structural and semantics of our data. Heterogeneous graph Neural Networks generally work by having separate non linear functions convolve over each edge type during message computation and over each node type when aggregating the learned information.

Loss Function

The Mean Squared Error(MSE) loss is the loss function used for our problem with the intention that the losses are minimized. We have adopted 2 methodologies to regularize the loss one each for the 2D and 3D KPI.

$$\text{MSE} = \frac{1}{N} \sum_{i=1}^N (y_i - \hat{y}_i)^2$$

$$\Delta \hat{y}_i = \hat{y}_{i+1} - \hat{y}_i$$

Loss Regularization for 2D KPI(Torque curve)

To smoothen out the curve for the 2D KPI(Torque curve) we apply a loss regularisation factor to take into account before backpropagating it the model during training. We observed the curve inherently follows a decreasing pattern and hence using this knowledge penalize the loss for non-decreasing values within each prediction.

$$\text{violations}_i = \text{ReLU}((\Delta \hat{y}_i)^2) = \text{ReLU}((\hat{y}_{i+1} - \hat{y}_i)^2)$$

$$\text{regularization_term} = \frac{1}{N-1} \sum_{i=1}^{N-1} \text{violations}_i$$

$$\text{Y1 Loss} = \text{MSE} + \lambda_{\text{regularisation_weight}} \times \text{regularized_term}$$

Theoretically, we would expect the model to generate better predictions but on closer observation we notice the curve is still not smooth. A reason to attribute this could be the model's incapability to infer that loss decrease depends on not just the prediction and target values but also within prediction values. Our deduction is that a single value calculated for the entire curve may not be sufficient to regularize this loss as we imagined.

Loss Customization for 3D KPI(Efficiency Grid)

The ETA Grid is a 3D plot of real numbers ranging between 0 and 100. We noticed in some portions of the grid, the plot not visible as it had nan values. As ANN cannot be trained to predict NAN values we had these replaced to -1 and have a binary mask constructed such that values corresponding to -1 in the target have value 0 and all other values as 1. The mask is then multiplied with both the target and the prediction. After this step, the MSE loss is calculated and backpropagated. Mathematically, this process can be expressed as follows:

$$\begin{aligned} \text{Let } M_{ij} &= \begin{cases} 1 & \text{if } y_{ij} \neq -1 \\ 0 & \text{if } y_{ij} = -1 \end{cases} \\ \hat{y}_{ij}^{\text{masked}} &= \hat{y}_{ij} \cdot M_{ij} \\ y_{ij}^{\text{masked}} &= y_{ij} \cdot M_{ij} \\ \text{Y2 Loss} &= \text{MSE}(\hat{y}^{\text{masked}}, y^{\text{masked}}) \end{aligned}$$

where M_{ij} : mask matrix

This formulation ensures that the -1 values (which replaced NaN values in the grid) are ignored in the loss calculation, as they are multiplied by 0 in the mask.

$$\text{Total Loss} = \text{Y1 Loss} + \text{Y2 Loss}$$

Evaluation Metrics

The evaluation metrics we have considered for our regression problem is the MSE loss and the R2 score. The model with the least loss and the highest R2 score is ideal for our application.

Experiments and Results

Firstly we train a MLP on the tabular representation of the data. We have used MinMax Scaler for normalization of both the input and targets. MSE loss function is used for the regression problem and Adam optimizer is used for optimization. We have the first results on the 2D KPI-Mgrenz prediction. These results are only considering about 400 examples. The dataset was split into training-validation(80:20) excluding test examples of about 19 examples.

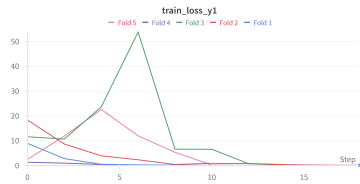


Figure 8: Training Loss for Torque Curve



Figure 9: Training Loss for ETA grid

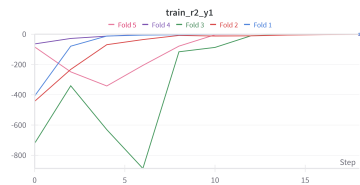


Figure 10: Training R2 Score for Torque Curve

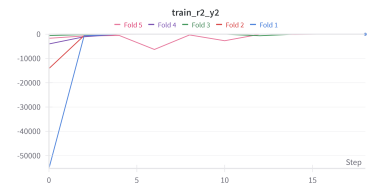


Figure 11: Training R2 Score for ETA grid

From the training plots we see that the model has converged after having run for 10 epochs with a learning rate of 0.0075.

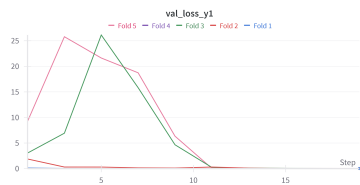


Figure 12: Validation Loss for Torque Curve

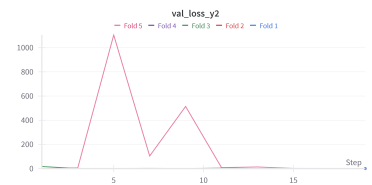


Figure 13: Validation Loss for ETA grid

The results of the MLP model from inference is as below:

We see a good fit of the model to the data with R2 score approaching close to 1. We have also enabled saving the trained model locally so it can be loaded on demand by the client to run inference.

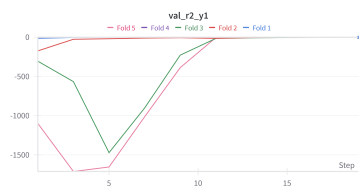


Figure 14: Validation R2 Score for Torque Curve

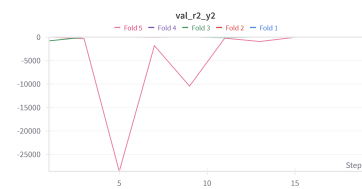
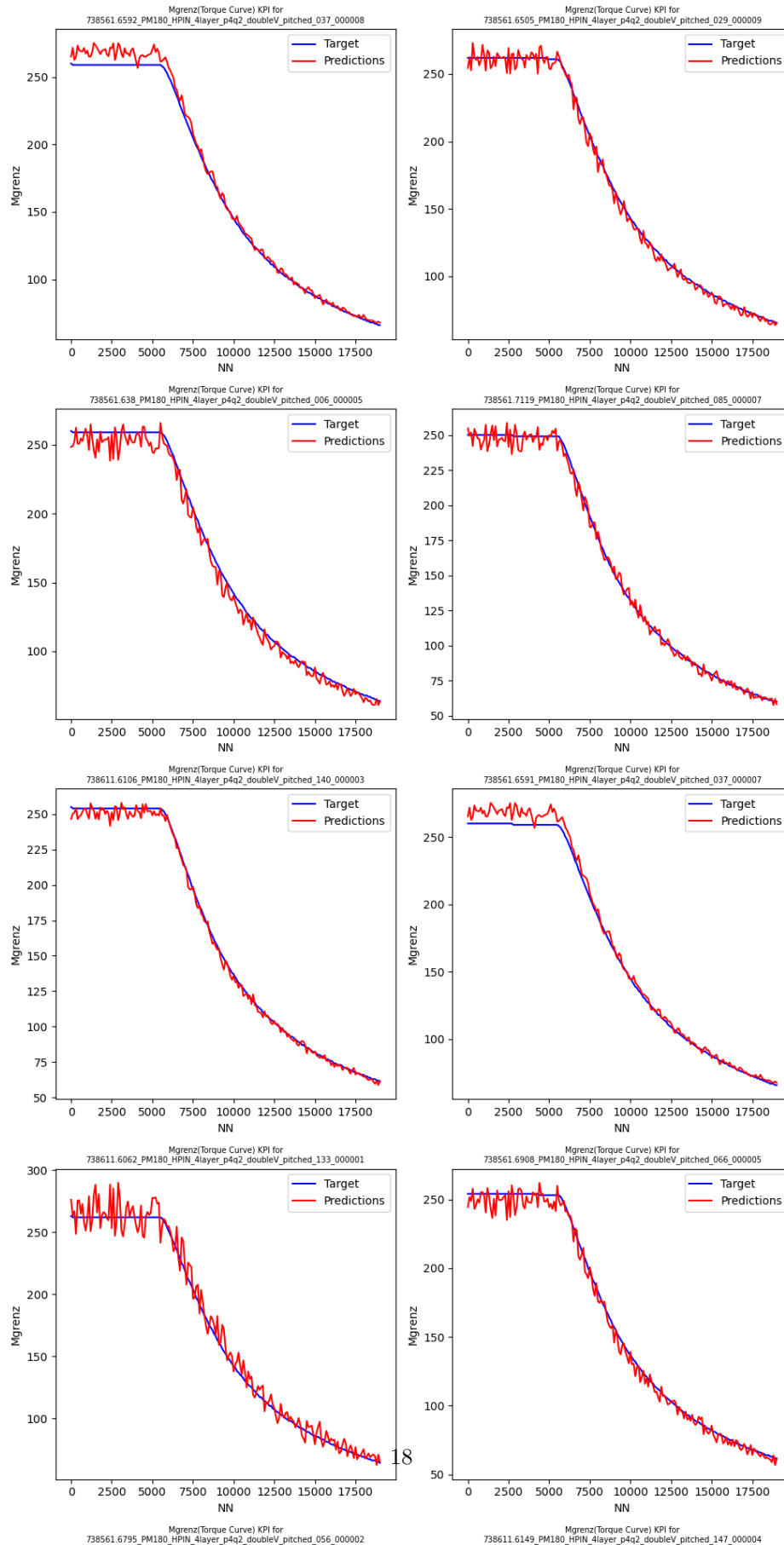


Figure 15: Validation R2 Score for ETA grid

Plots of 25 to 35 Examples from the Test Dataset



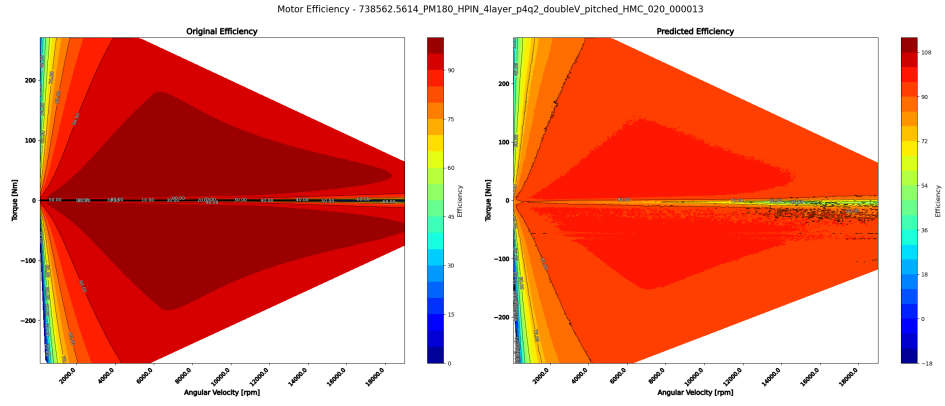


Figure 17: 1st MLP Training Results for 3D KPI(ETA)

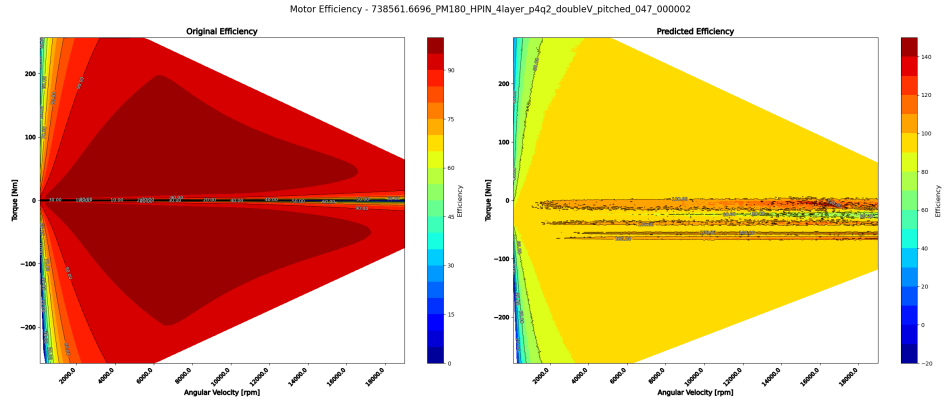


Figure 18: 2nd MLP Training Results for 3D KPI(ETA)

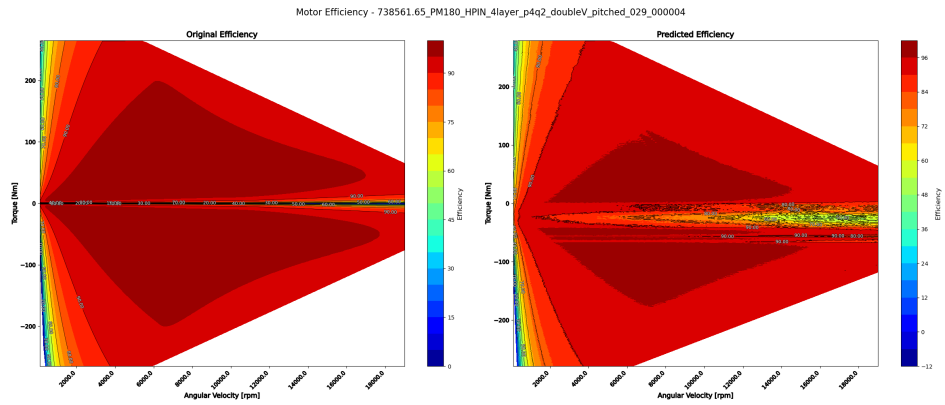


Figure 19: 3rd MLP Training Results for 3D KPI(ETA)

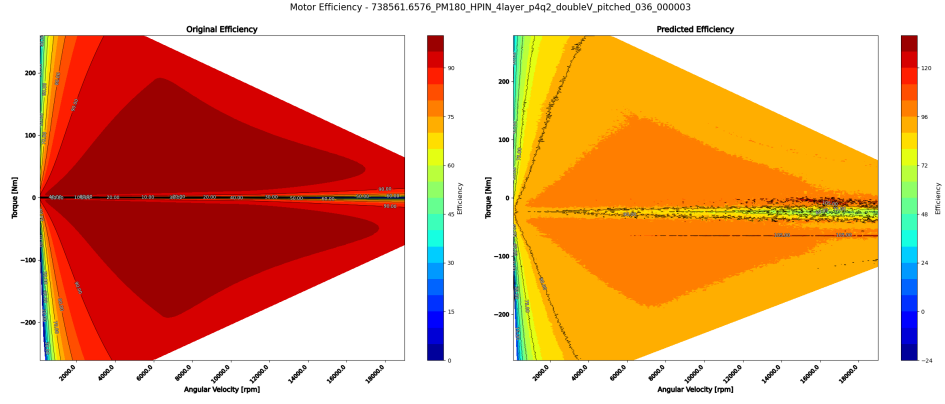


Figure 20: 4th MLP Training Results for 3D KPI(ETA)

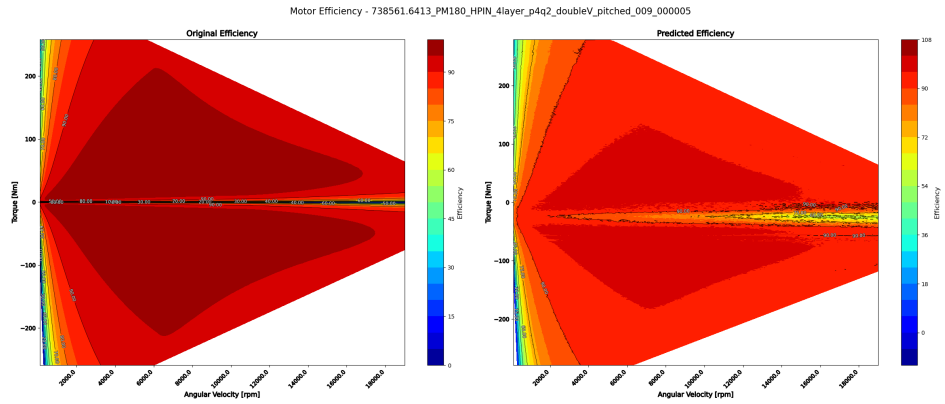


Figure 21: 5th MLP Training Results for 3D KPI(ETA)

Conclusion

List of Figures

1	V1 Magnet (Source:Valeo)	6
2	V2 Magnet (Source:Valeo)	6
3	Nabla Magnet (Source:Valeo)	6
4	EM Design Flowchart	7
5	Complete EM Geometry(Source:Valeo)	9
6	1/8 Motor Crossection	10
7	HetGraph	12
8	Training Loss for Torque Curve	16
9	Training Loss for ETA grid	16
10	Training R2 Score for Torque Curve	16
11	Training R2 Score for ETA grid	16
12	Validation Loss for Torque Curve	16
13	Validation Loss for ETA grid	16
14	Validation R2 Score for Torque Curve	17
15	Validation R2 Score for ETA grid	17
16	MLP Training Results for 2D KPI(Mgrenz)	18
17	1st MLP Training Results for 3D KPI(ETA)	19
18	2nd MLP Training Results for 3D KPI(ETA)	19
19	3rd MLP Training Results for 3D KPI(ETA)	19
20	4th MLP Training Results for 3D KPI(ETA)	20
21	5th MLP Training Results for 3D KPI(ETA)	20

List of Tables

Appendix

Bibliography

Declaration on oath

I hereby certify that I have written my master thesis independently and have not yet submitted it for examination purposes elsewhere. All sources and aids used are listed, literal and meaningful quotations have been marked as such.

Lilly Abraham K64889, 11.12.2024

Consent to Plagiarism Check

I hereby agree that my submitted work may be sent to PlagScan (www.plagscan.com) in digital form for the purpose of checking for plagiarism and that it may be temporarily (max. 5 years) stored in the database maintained by PlagScan as well as personal data which are part of this work may be stored there.

Consent is voluntary. Without this consent, the plagiarism check cannot be prevented by removing all personal data and protecting the copyright requirements. Consent to the storage and use of personal data may be revoked at any time by notifying the faculty.

Lilly Abraham K64889, 11.12.2024