Technical University of Applied Sciences Würzburg-Schweinfurt (THWS)

Faculty of Computer Science and Business Information Systems

Master Thesis

Electrical Engine Efficiency Prediction Bypassing PDE Simulator

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 Key Performance Indicators(KPI)s of topology invariant Interior Permanent Magnet Synchronous Magnets (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 Graph Neural Network(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 hyper-parameter tuning settings and as a baseline with an Multi Layer Perceptron(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 Efficieny 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. Her dedication and commitment to our work has been inspiring to me especially on how we transformed statistical math into modelling that I might have developed a new love for academia.
I would also like to express my sincere gratitude to Valeo for providing us with the dataset. Special thanks are in order to Daniel and Leo for sharing valuable insights of the data from an electromechanical standpoint and for giving me a detailed understanding of my task.
I owe my Mother and my Siblings my accomplishments. They have been very instrumental in making it possible for me to pursue a Master's degree outside my home country and for the endless support thoughout.
Finally, I humble myself before God Almighty for all his blessings and for giving me the strength to persevere and bring my dreams to fruition.

Contents

Abbreviations

GNN Graph Neural Network

MLP Multi Linear Perceptron

KPI Key Performance Indicator

EM Electric Motor

FEM Finite Element Method

CNN Convolution Neural Network

2D 2 Dimension

3D 3 Dimension

MSE Mean Squared Error

RMSE Root Mean Squared Error

NaN Not a Number

ReLU Rectified Linear Unit

Chapter 1

Introduction

In the design of electric motors, vast amounts of data are generated to determine which design of an Electric Motor (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 Finite Element Method (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:

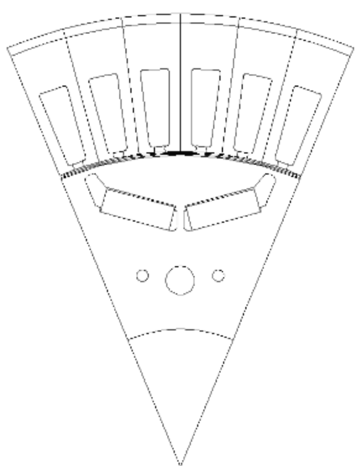

Figure 1.1: V1 Magnet (Source: Valeo)

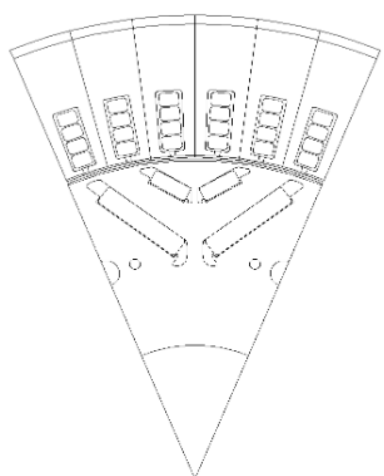

Figure 1.2: V2 Magnet (Source:Valeo)

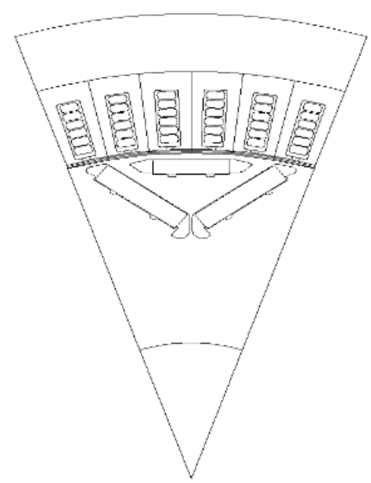

Figure 1.3: Nabla Magnet (Source:Valeo)

The current approach to predict the KPIs of different EM design variants is to create a design mesh from the paramteric description of the motor with Matlab. Multiple FEM simulations is done on this mesh which is then post processed and the intermediary outputs are forwarded to the Motor Builder. Several Motor builder settings are then adjusted to get the plots of the desired KPIs.
This master thesis explores a way to do surrogate modelling of the current process as is highlighted in Figure 1.4 by making use of GNN or MLP for the modelling of electrical engine designs described parameterically.

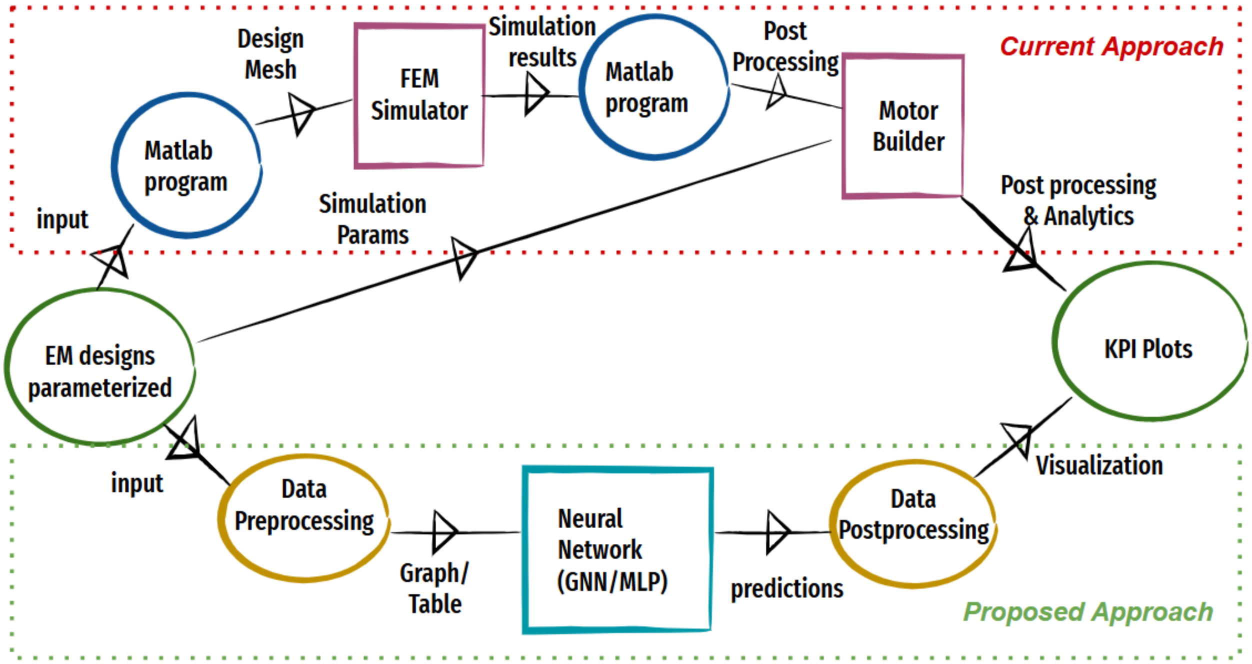

Figure 1.4: EM Design Flowchart

1.1 Objective

Our task is to predict 2 KPIs namely the Torque curve and the ETA grid from the parameteric description of topology invariant Electris Motors.
The Torque curve is a 2D plot ie, a vector across speed ranges and the Efficiency grid is a 3D plot ie, a matrix of dimensions torque ranges times speed ranges.
Both the outputs are continuous values which makes it a regression problem.

1.2 Problem Statement

1. We have a regression problem but as we are predicting two values it makes it a multi regression problem.
2. Another problem we came across is the Torque Curve typically harbours integers values however it is a regression problem and needs to be modelled as float values.
3. The Efficiency grid dimensions vary across EM variants, and we need the target from the model to be of a fixed size. This problem is mitigated in 3.1.3
4. The ranges among the 1st and 2nd target vary significantly and is yet another problem we overcome in 3.2
5. Furthermore, we presume graph representation of the data will be more logical than tabular representation due to its ability to grasp the connections within the EM structural properties better. This would be even more realistic with achieving topology invariance than the tabular representation and conserve memory and compute in the long run. However, we realize that our problem cannot be solved using Homogeneous GNN which is relatively simpler and is built on a single node and edge type. In order to model our problem as a graph, we need to represent it as a Heterogeneous graph.

1.3 Research Question

GNN in general have not been to less explored even so more the heterogeneous GNN. Particularly in the scenario of Electric Motor Modelling, there has been no publications with GNNs. Hence the need to check its feasibility and its performance with our benchmarks on tabular data. Additionally, existing Heterogeneous GNNs works e.g on recommendation networks, academic networks, information networks, social networks etc involves one large graph with multiple node and edge types. However, our problem involves creation of multiple heterogeneous graphs ie, 1 per EM variant.
Therefore, the applicability of Heterogeneous GNNs for our problem is to be seen.

Due to the challenge of predicting ETA grid of varying dimensions for each motor variant, we decided against building a model architecture wherein the 2nd KPI is dependent on the 1st KPI.
This in a way could be overcome if we build 2 models one for the 2D and 3D KPIs and thus feed in the corresponding 2D predictions when training the latter.
However, it would be computationally expensive and does not help in the scenario when we might need to generate EM parameteric descriptions. Additionally we deemed it unnecessary as the dimensions of the ETA grid vary with the torque curve and not necessarily the ETA values.

1.4 Thesis Structure

Over the course of the thesis we shall refer the Torque curve as Mgrenz KPI and the Efficiency grid as ETA KPI respectively. The thesis is structured to follow sections namely Literature Review, Dataset, Modelling, Experiments and Results, Conclusion, and Bibliography.
In Literature Review section will introduce the works that has already been carried out in this domain.
In the Dataset section a detailed insight to how our data is structured is elaborated.
In the Modelling section, we introduce the methodologies used to tackle the problem.
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 glimpse into areas of improvement.
Finally the Bibliography section lists out the articles cited for this thesis.

Chapter 2

Literature Review

There has been extensive research in modeling the Electric Motor with Convolution Neural Network (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.
Reproducing images is not known to be the best approach given the infamous known fact that AI generated images are faulty. However 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 GNNs 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.

Chapter 3

Dataset

Valeo an automotive company has supplied the dataset consisting of close to 1500 Double V Electric Motor parameters. Around 89 parameters which comprises of the geometric, physical and simulation properties of the motor are chosen among the 196 parameters depending on its overall variability and significance.

Figure 3.1 shows the geometry of a whole Double V motor

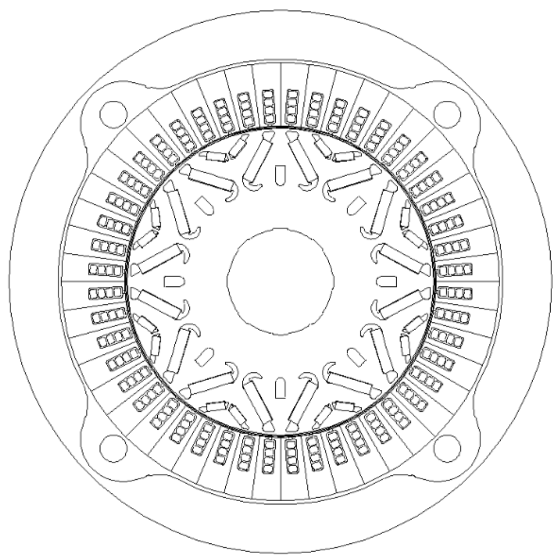

Figure 3.1: Complete EM Geometry(Source:Valeo)

Figure 3.2 outlines the geometry of 1/8 cross-section of the same motor.

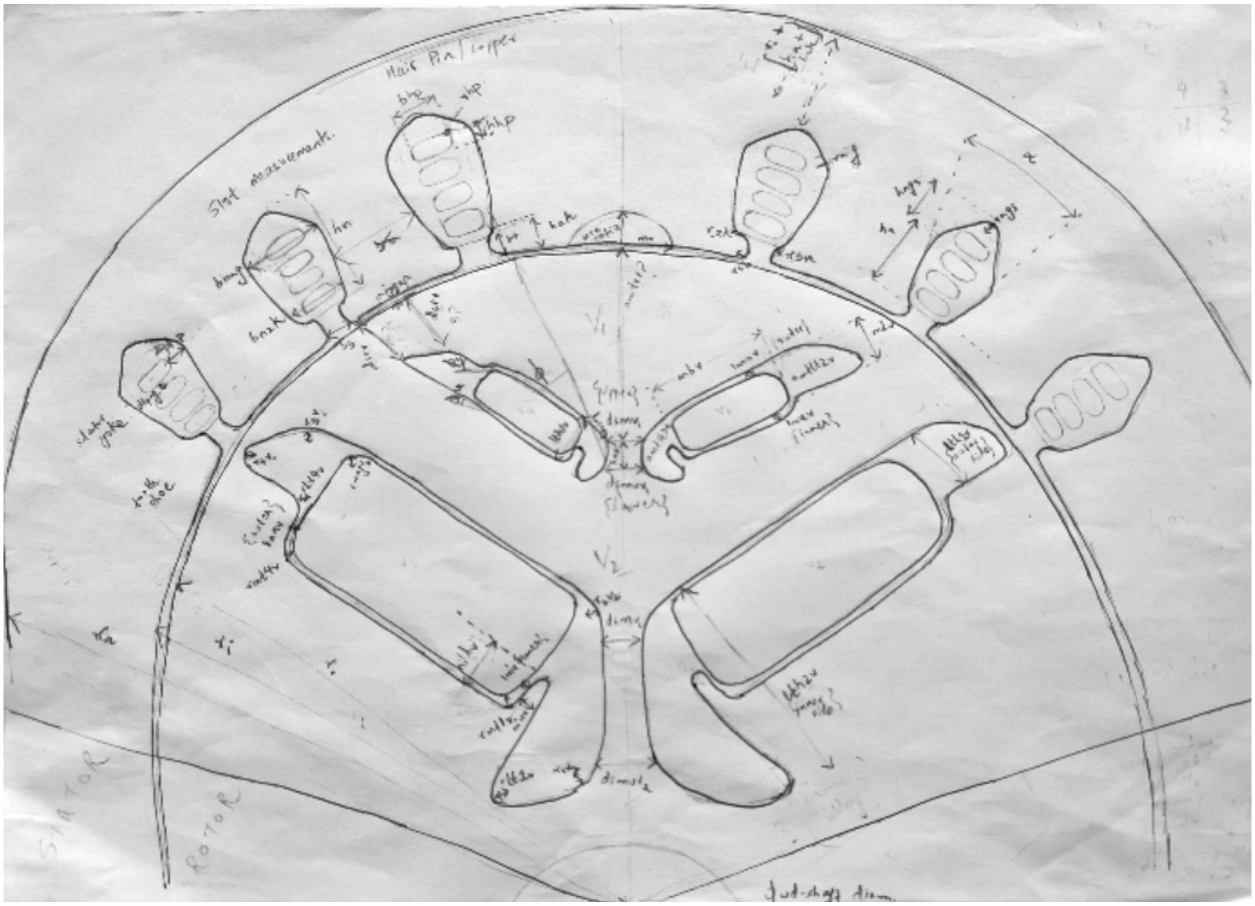

Figure 3.2: 1/8 Motor Crossection

Valeo has shared approximately 1500 Excel Workbook files for each motor variant. Each of the excel files contain multiple sheets. The sheets of interest to us are as below :

Sheet	Sheet Name	Description
Motor Parameters	input_data	• Contains the input parameters for the motor model. • Includes geometric, physical, and simulation properties. • Unit dimensions if applicable are generally mm or degrees
Speed Grid	NN	• Contains the input parameters for the motor model. • Includes geometric, physical, and simulation properties. • Unit dimensions if applicable are generally mm or degrees
Torque Grid	MM	• Contains the torque grid. • Used for plotting the ETA KPI.
ETA Grid	ETA	• Contains the ETA KPI. • Have the same dimensions as that of NN and MM sheets
Torque Curve	Mgrenz	• Contains the values corresponding to Mgrenz KPI. • Have the same columns corresponding to the speed grid.

Table 3.1: Excel File Structure of an EM variant

3.1 Data Preprocessing for MLP

For modelling the MLP, we present the data in tabular form with the parameters corresponding to columns. In order to make the data compatible with our model, some level of data processing was carried out as elaborated below.

3.1.1 Data Exploration of the Input Parameters

All parameters including the additional ones in each topology are considered as a separate columns and therefore if a particular column is topology dependent the data of the other topologies for that corresponding column is treated as 0 values.
The values are read and stored in their float equivalent to preserve data precision. Furthermore all degree columns are converted to their equivalent radian values to be fed to the model..EXPLAIN THE REASON WHY WITH CITATION PROBABLY::::

Parameter	Unit of Dimension	Mean	Standard Deviation	Value Range	Single V Magnet Topology	Double V Magnet Topology	Nabla Magnet Topology
General Parameters							
N	#	4.0	0.0	4.0–4.0	✓	✓	✓
simQ	#	6.0	0.0	6.0–6.0	✓	✓	✓
r_a	mm	9.000000e-02	2.554375e-15	9.000000e-02–9.000000e-02	✓	✓	✓
r_i	mm	0.064433	0.000902	0.064000–0.067000	✓	✓	✓
Rotor Parameters							
rad_phiv2	rad	-0.511319	0.063678	-0.645772–0.000000	×	✓	×
lmsov2	mm	-0.000289	0.000035	-0.000300–0.000000	×	✓	×
lth1v2	mm	0.005395	0.000359	0.000000–0.005450	×	✓	×
lth2v2	mm	0.002789	0.000178	0.000000–0.002800	×	✓	×
r1v2	mm	0.002097	0.000322	0.000000–0.002200	×	✓	×
r11v2	mm	0.000326	0.000040	0.000000–0.000600	×	✓	×
r2v2	mm	0.001873	0.000133	0.000000–0.001900	×	✓	×
r3v2	mm	0.000697	0.000044	0.000000–0.000700	×	✓	×
r4v2	mm	0.000747	0.000048	0.000000–0.000750	×	✓	×
rmt1v2	mm	0.000249	0.000016	0.000000–0.000250	×	✓	×
rmt4v2	mm	0.000249	0.000016	0.000000–0.000250	×	✓	×
rlt1v2	mm	0.000185	0.000045	0.000000–0.000200	×	✓	×
rlt4v2	mm	0.000249	0.000016	0.000000–0.000250	×	✓	×
hav2	mm	0.004942	0.000336	0.000000–0.005000	×	✓	×
mbv2	mm	0.017706	0.001175	0.000000–0.018100	×	✓	×
mhv2	mm	0.003649	0.000265	0.000000–0.003800	×	✓	×
rmagv2	mm	0.000498	0.000032	0.000000–0.000500	×	✓	×
dsmv2	mm	0.002925	0.000194	0.000000–0.003100	×	✓	×
dsmuv2	mm	0.002925	0.000194	0.000000–0.003100	×	✓	×
amtrv2	mm	0.015888	0.001024	0.000000–0.016000	×	✓	×
dsrv2	mm	0.000996	0.000064	0.000000–0.001000	×	✓	×
lmav2	mm	0.00010	0.00003	0.00000–0.00011	×	✓	×
lmiv2	mm	0.000109	0.000008	0.000000–0.000110	×	✓	×
lmov2	mm	0.000055	0.000015	0.000000–0.000100	×	✓	×
lmuv2	mm	0.000145	10.000017	0.000000–0.000150	×	✓	×
rad_phiv1	rad	-0.694266	0.054700	-0.785398—0.453786	✓	✓	✓
lmsov1	mm	-0.000501	0.000094	-0.000530–0.000500	✓	✓	✓
lth1v1	mm	0.002889	0.000178	0.002855–0.005450	✓	✓	✓
lth2v1	mm	0.002104	0.000059	0.002100–0.003200	✓	✓	✓
r1v1	mm	0.000407	0.000108	0.000400–0.002200	✓	✓	✓
r11v1	mm	0.000219	0.000045	0.000100–0.000600	✓	✓	✓
r2v1	mm	0.000216	0.000100	0.000200–0.001900	✓	✓	✓
r3v1	mm	0.000899	0.000013	0.000700–0.000900	✓	✓	✓
r4v1	mm	0.000501	0.000016	0.000500–0.000750	✓	✓	✓
rmt1v1	mm	2.500000e-04	8.839232e-18	2.500000e-04–2.500000e-04	✓	✓	✓
rmt4v1	mm	2.500000e-04	8.839232e-18	2.500000e-04–2.500000e-04	✓	✓	✓
rlt1v1	mm	0.000117	0.000056	0.000050–0.000250	✓	✓	✓

Continued on next page

Table 3.2 – continued from previous page

Parameter	**Unit of Dimension**	**Mean**	**Standard Deviation**	**Value Range**	**Single V Magnet Topology**	**Double V Magnet Topology**	**Nabla Magnet Topology**
rlt4v1	mm	2.500000e-04	8.839232e-18	2.500000e-04–2.500000e-04	✓	✓	✓
hav1	mm	0.002918	0.000136	0.002900–0.005000	✓	✓	✓
mbv1	mm	0.007643	0.000588	0.007500–0.018150	✓	✓	✓
mhv1	mm	0.002808	0.000140	0.002700–0.005000	✓	✓	✓
rmagv1	mm	5.000000e-04	1.767846e-17	5.000000e-04–5.000000e-04	✓	✓	✓
dsmv1	mm	0.001079	0.000141	0.000800–0.002800	✓	✓	✓
dsmuv1	mm	0.001079	0.000147	0.000800–0.002920	✓	✓	✓
amtrv1	mm	0.005538	0.000634	0.005500–0.019000	✓	✓	✓
dsrv1	mm	0.000752	0.000032	0.000750–0.001250	✓	✓	✓
lmav1	mm	0.000092	0.000026	0.000010–0.000110	✓	✓	✓
lmiv1	mm	1.000405e-04	6.354235e-07	1.000000e-04–1.100000e-04	✓	✓	✓
lmov1	mm	0.000055	0.000015	0.000050–0.000100	✓	✓	✓
lmuv1	mm	0.000145	0.000015	0.000100–0.000150	✓	✓	✓
rad_phi3b1	rad	-0.002522	0.056001	-1.256637–0.000000	×	×	✓
rad_phi4b1	rad	-0.000530	0.011775	-0.261799–0.000000	×	×	✓
lmsob1	mm	0.000002	0.000034	0.000000–0.000750	×	×	✓
lthb1	mm	0.000006	0.000128	0.000000–0.002900	×	×	✓
r2b1	mm	0.000002	0.000045	0.000000–0.001000	×	×	✓
r3b1	mm	0.000002	0.000034	0.000000–0.001000	×	×	✓
r4b1	mm	5.064146e-07	1.124422e-05	0.000000e+00–2.500000e-04	×	×	✓
r5b1	mm	5.064146e-07	1.124422e-05	0.000000e+00–2.500000e-04	×	×	✓
lgr3b1	mm	0.000001	0.000022	0.000000–0.000500	×	×	✓
lgr4b1	mm	6.076975e-07	1.349307e-05	0.000000e+00–3.000000e-04	×	×	✓
mbb1	mm	0.000030	0.000675	0.000000–0.015000	×	×	✓
mhb1	mm	0.000006	0.000144	0.000000–0.003200	×	×	✓
mtbb1	mm	0.000030	0.000675	0.000000–0.015000	×	×	✓
rmagb1	mm	0.000001	0.000022	0.000000–0.000500	×	×	✓
amtrb1	mm	0.000004	0.000098	0.000000–0.002500	×	×	✓
dsr3b1	mm	0.000003	0.000066	0.000000–0.001850	×	×	✓
dsr4b1	mm	0.000004	0.000083	0.000000–0.001850	×	×	✓
lmob1	mm	2.025658e-07	4.497689e-06	0.000000e+00–1.000000e-04	×	×	✓
lmub1	mm	3.038488e-07	6.746534e-06	0.000000e+00–1.500000e-04	×	×	✓
lmsub1	mm	0.000004	0.000081	0.000000–0.001800	×	×	✓
Stator Parameters							
airgap	mm	1.000000e-03	3.535693e-17	1.000000e-03–1.000000e-03	✓	✓	✓
b_nng	mm	0.005049	0.000091	0.004646–0.005120	✓	✓	✓
b_nzk	mm	0.004557	0.000057	0.004450–0.004646	✓	✓	✓

Continued on next page

Table 3.2 – continued from previous page

Parameter	Unit of Dimension	Mean	Standard Deviation	Value Range	Single V Magnet Topology	Double V Magnet Topology	Nabla Magnet Topology
b_s	mm	0.001002	0.000025	0.001000–0.001400	✓	✓	✓
h_n	mm	0.011149	0.000806	0.009200–0.013939	✓	✓	✓
h_s	mm	1.000000e-03	3.535693e-17	1.000000e-03–1.000000e-03	✓	✓	✓
r_sn	mm	2.500000e-04	8.839232e-18	2.500000e-04–2.500000e-04	✓	✓	✓
r_zk	mm	0.000501	0.000019	0.000500–0.000800	✓	✓	✓
r_ng	mm	0.000501	0.000019	0.000500–0.000800	✓	✓	✓
h_zk	mm	1.000000e-03	3.535693e-17	1.000000e-03–1.000000e-03	✓	✓	✓
bhp	mm	0.003736	0.000081	0.003550–0.003800	✓	✓	✓
hhp	mm	0.002386	0.000199	0.001900–0.002840	✓	✓	✓
rhp	mm	0.000636	0.000045	0.000500–0.000800	✓	✓	✓
dhphp	mm	0.000264	0.000014	0.000263–0.000485	✓	✓	✓
dhpng	mm	0.000406	0.000003	0.000406–0.000453	✓	✓	✓

Table 3.2: EM Input Parameters

3.1.2 Data Exploration of the Mgrenz KPI(Torque Curve)

Figure 3.3 shows the standard deviation of few samples of the Mgrenz KPI.

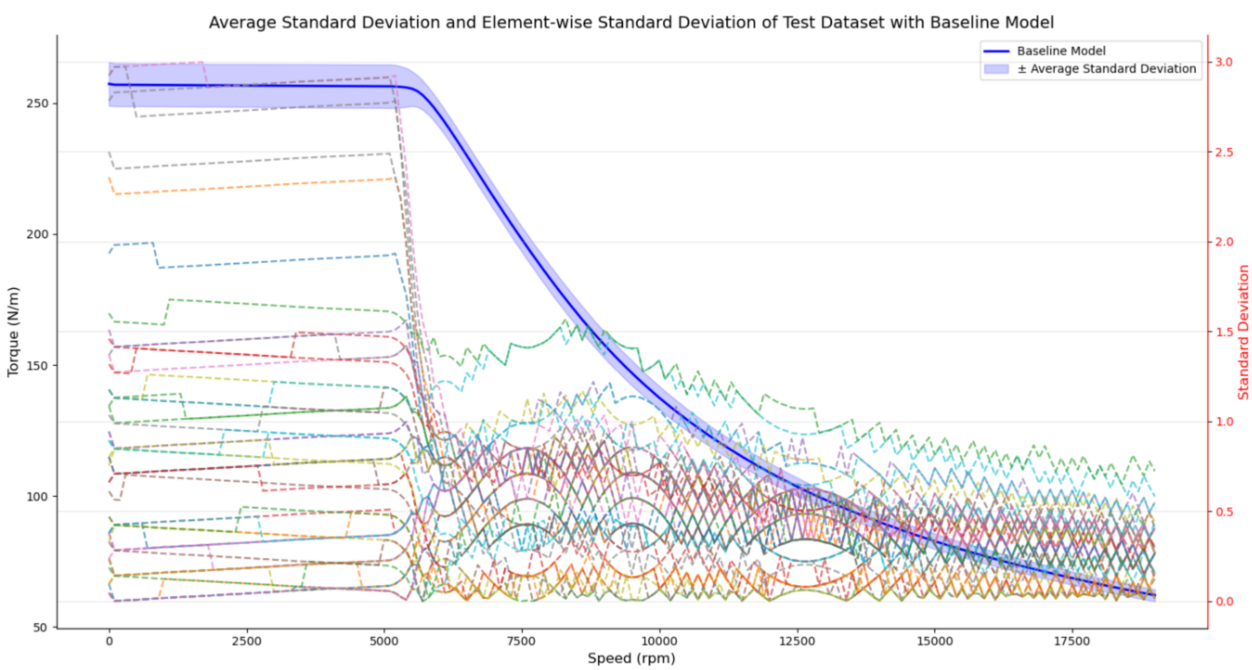

Figure 3.3: Standard Deviation of 2D KPI(ETA) Targets

We make the below 2 observations from it :

1. The Root Mean Squared Error (RMSE) is at its peak at low speeds.
2. The curve to an extent resembles a mirrored S shape. This finding is critical for how we modelled the loss regularization for the target and will be further elaborated in 5.2.1.

3.1.3 Data Exploration of the ETA KPI(Efficiency Grid)

As the target values Mgrenz KPI and ETA KPI are not provided with the correct dimensions we have an additional step which takes the maximum torque value from the Mgrenz KPI and create a similar grid ranging from -maximum torque to maximum torque.
We then choose only the rows corresponding to this range from the actual MM grid supplied and the same row indices is used to retrieve the ETA KPI.
This step ensures that we grant the model the correct dimensions of the ETA KPI based on Torque KPI and predict likewise.

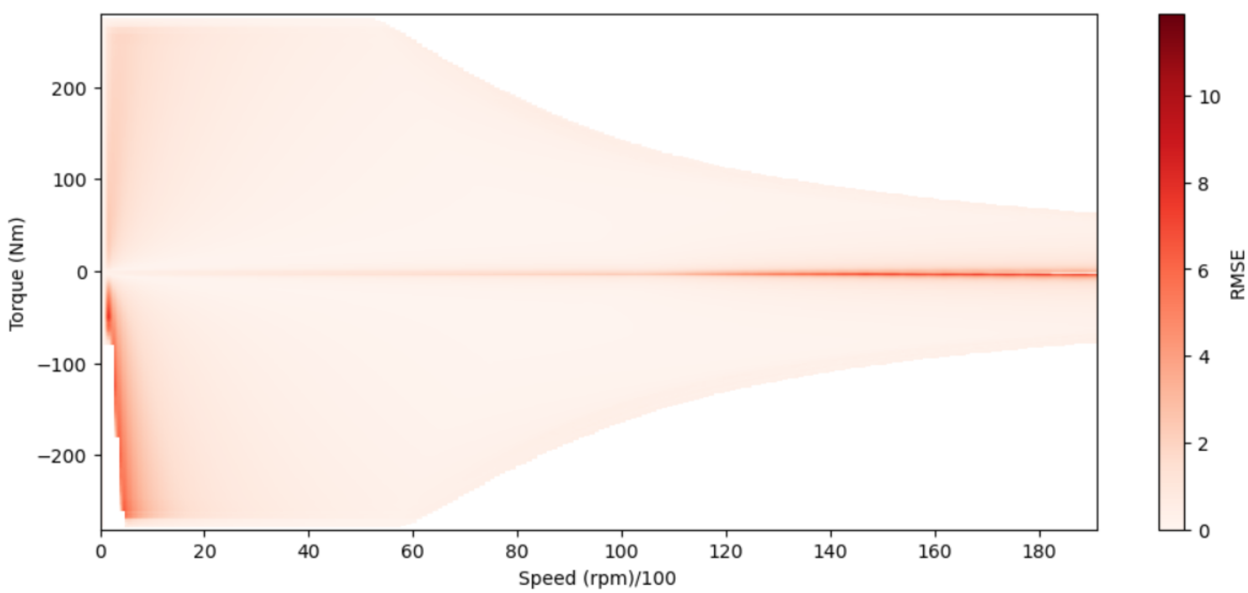

Figure 3.4: Standard Deviation of ETA KPI

From our analysis of the above plots on the standard deviation of the ETA KPI from the test dataset, we notice that the data shared was faulty with respecet to the efficiency values corresponding to negative torque values. Since these are FEM simulations, it is probably an effect of a post processing step taken by the Motor builder. The efficiency values for negative torque values correspond to when motor is in generating mode is not the replica as how it is for positive torque values when motor is in monitoring mode. This is evident from low speed-high torque distribution area where we can see Not a Number (NaN) values across the test dataset

This observation made us decide on dropping the negative ETA KPI and only predict the positive ETA KPI.

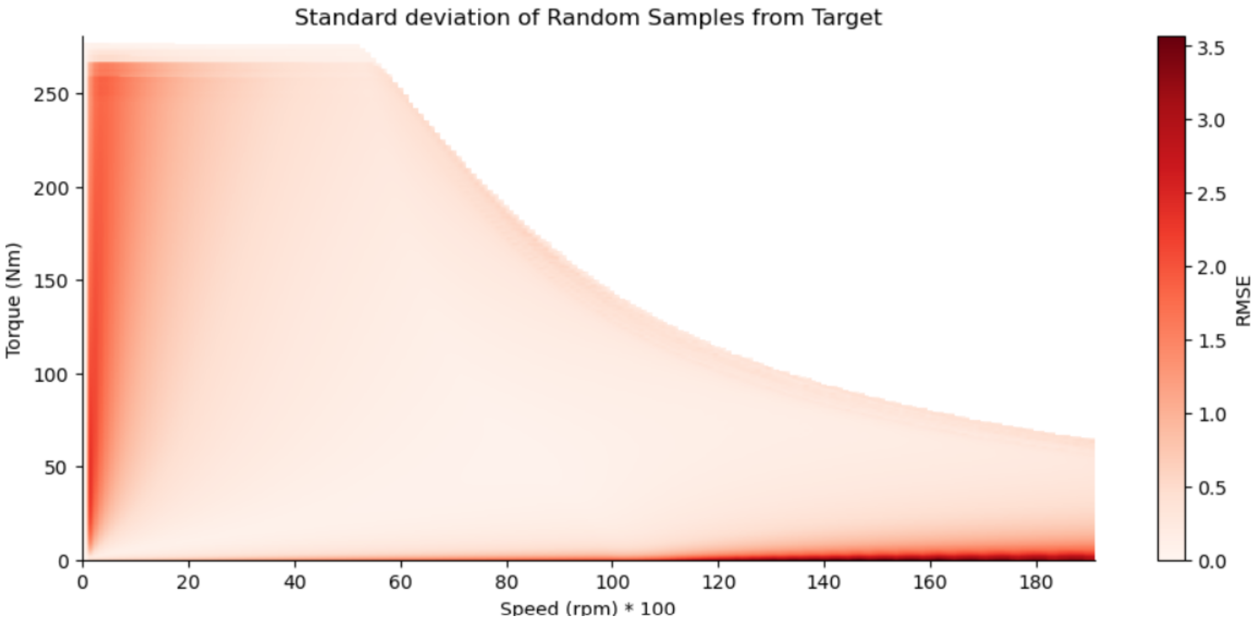

Figure 3.5: Standard Deviation of ETA KPI Positive Grid

On closer inspection, we can observe the deviation is at its peak at low torques, low speeds and the border of the curve within the grid. We discuss the modelling of this information in 5.2.2.

Additionally the ETA KPI envelope is completely dependent on its equivalent Mgrenz KPI curve. The area beneath the boundary of which is looked into by the EM manufacturers to determine the car's efficiency in the operating cycle. This is yet another finding we use in Post Processing as is further elaborated in 5.5 To conclude, we have decided to only consider the ETA KPI in monitoring mode for predictions. It can be mirrored to replicate the efficiency when it is in generating mode.

Figure 3.6: Standard Deviation of ETA KPI Positive Grid across Speed Intervals

Figure 3.6 gives us the big picture of how the efficiency values are distributed across equally spaced intervals of speed.

Due to the infamous fact that reading sheets in excel files take up a lot of time and compute, we read the files as a onetime job when creating the table and store them into pythonic objects for faster access for training.
Both the input and target values for the Mgrenz KPI is stored locally as csv files whereas those of the ETA KPI is stored as separate csv files per variant considering it is in the form of a 2 Dimension (2D) array.
The csv files are then concatenated and stored into a numpy array conserving dimensionality by padding NaN values to match dimensionality of the grid corresponding to the Mgrenz KPI with the largest torque value.
In our case the value is 280 but this is subject to change as we receive more data and can be overridden by the user.
The array is then saved locally as a .npy file for easy access and loading during training.

The input data consists of about 1500 examples mostly from the Double V Magnet Topology (and about 3 examples each for the other 2 topologies).

3.2 Scaling

We have used Standard Scaling for the input and both the outputs.
The Scaling is formulated as shown below :

$$\mathrm{z} = \frac{x-\mu}{\sigma} \tag{3.1}$$

where

- x : *Input*
- μ : *Mean*
- σ : *Standard Deviation*

For the Input both Mean and Standard deviation are calculated across columns. This is attributed to the fact we have columns with different ranges for the input. Meanwhile the torque curve and the ETA grid are flattened to be 1D and thus have only 1 column.

We then removed scaling for the targets because these donot enter the model but is only used for loss calculation. With lower learning rates and higher epochs, we see the model's y1 performing better than with scaling. A plausible reason could be since the y1 targets are integers typically but we use torch tensor as it is a regression problem. This misleads the model with scaled targets to lose precision by generating within the full capacity of float tensors. However without scaling, we observe the model spits out integers itself even when the datatype is still torch tensor. This inturn fixes the y1 earlier problem of having fluctuations in the curve. Furthermore it now eliminates the need for regularizing the Y1 targets as predictions are as per target a smooth decreasing curve.

Additionally, the NaN values in y2 are handled by masking before loss calculation as is defended in Section CITE Loss function Initially before masking y2's NaN we try to set it as an incredibly high value - but this resulted in poor predictions as the model must have been confused and tried to increase its spread of predictions to cover this value and so all true values were also predicted to be close to this dummy value–with scaling

3.3 Dataset splitting

We will be using Pytorch Machine Learning library in Python and therefore convert the data to float tensors and collate them into a Tensor Dataset. 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.
The reason we have a separate test dataset from the validation is to ensure that there is no data leakage as we donot want to overfit the testdataset with the hyperparameters we choose during training.

Within 5 folds, we expect to cover most grounds on training and have good monitoring on the model's performance for each fold. We have also used Dataloaders to split the dataset into batches that fits into our GPU memory.

Chapter 4

Graph Modelling

We intended to model our problem as a Graph and solve using Graph Neural Networks. We presume this will be a more clever way of representing our usecase in comparision to tabular data.. The idea was to make use of the Graph dynamics in our usecase and aggregate features that are semantically similar.

GNNs are primarily designed for homogeneous graphs associated with a single type of nodes and edges, following a neighborhood aggregation scheme to capture structural information of a graph, where the representation of each node is computed by recursively aggregating the features of neighbor node [**?**].

These relations can be composited with each other to form high-level semantic relations, which are represented as metapath [**?**].

Heterogeneous graphs contain more comprehensive information and rich semantics and require specifically designed models [**?**].

Heterogeneous graph neural networks (HGNNs) have powerful capability to embed rich structural and semantic information of a heterogeneous graph into node representations. [**?**]

The main method of aggregating information within the nodes of a Graph is through Message Passing. Graph Neural Networks can be broadly classified as : Homogeneous GNN Here the nodes and edges are of the same type. Message passing is done across the neighbouring nodes and edges over hops until it learns a representation equivalent from its neighbours Heterogeneous GNN Wherein the node and edge are of different types. Here message passing is conditioned on the node and edge type thus allowing the flow of information to be more controlled.

Standard Message Passing GNNs (MP-GNNs) can not trivially be applied to heterogeneous graph data, as node and edge features from different types can not be processed by the same functions due to differences in feature type. A natural way to circumvent this is to implement message and update functions individually for each edge type. During runtime, the MP-GNN algorithm would need to iterate over edge type dictionaries during message computation and over node type dictionaries during node updates.

4.1 Heterogeneous GNN 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.
This property is crucial in modelling our use case as we will then have similar node-edge types per topology. In addition the count of certain parameters with the motor such as stator poles with its corresponding slot and rotor magnets is made more comprehendable to the model having new nodes and edges whereas for the MLP architecture this information is represented only as a number in yet another column.
Heterogeneous GNN 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.

Inspired by the promising advantages of HGNN, we took the effort of modelling one for only the Double V Magnet Topology

Chapter 5

Modelling & Evaluation

Since we aim to predict continuous vector values, we model this task into a regression problem. Additionally since we predict 2 targets of continouous values, we make it a more complicated multi-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.

5.1 MLP Model

For the MLP model, we use a single model with input features corresponding to all the features in the tabular topology invariant representation of the data.
The model architecture is build to predict both the Mgrenz KPI and ETA KPIs by having 2 separate output layers for each of the KPIs.
Since the Mgrenz KPI's targets are relatively learnable than that of the ETA KPI's targets we have experimented with fewer feed forward layers in the former than in the latter.
We have a hyperparameter to control the number of neurons in each hidden layer this can be tuned and is further discussed in 6.1.
Rectified Linear Unit (ReLU) layers were also added in between to serve as the activation function and produce non-linearities and so noise in the network.
Dropout layers ensure that not all neurons in each layer are used up during training to prevent the model from memorizing the data and hence overfitting. We have a hyperparameter to control the dropout rate at which we freeze the neurons when training also to be discussed in 6.1.
Batch normalisation layers are used to normalize the input from the ReLU activations applied on it and so mitigate internal covariate shift to the next layer and hence speed up the training process.
Thus both batch normalisation and dropout layers stabilize the network training.

Figure 5.1 gives an outline on how the MLP Model architecture is designed.

1. Input
   The input layer takes in all features of the tabular data which is 89 in our case as scaled tensors.

2. MLP Shared
   The MLP Shared block is a sequential block comprising of 2 Linear Layers with the input features and neurons of each hidden layer to be a learnable size we tune. We donot increase the number of neurons in the hidden layers within this block as it needs to be in the range of input features and output features(in this case Mgrenz KPI) as well as in the multiples of 8 as GPUs are most optimized for the latter. Furthermore we have Batch Normalisation layers between each linear and ReLU activation function in addition to drop out layers. The 2 Linear Layers enable the network at the start to learn a rich represntation of the data at the intial feature extraction phase.

3. MLP Mgrenz
   This block comprises of Sequential 1 Linear Layer with the output feature to be the size of the Mgrenz

KPI and a ReLU activation function. We use a ReLU activation function at the end of the output layer as the targets are inherently always positive values and this encourages the model to adhere to this fact.

4. MLP ETA
This block comprises of Sequential 3 Linear Layer with the output feature to be the size of the ETA KPI. Here we also increase the neurons of the hidden layers as we are not limited by the dimensionality of the output feature. This would enable the model to be more strong and grasp the complex patterns in the data better. As usual we have batch normalisation, dropout and ReLU activation functions between the 1st two Linear layers. For the Last Linear layer we have the output features corresponding to the target dimensions and a ReLU activation again as the targets are inherently always positive values and this encourages the model to adhere to this fact.

5. Mgrenz KPI
The number of output features correspond to the target size 191. Although the targets for the Mgrenz KPI are an array of integer values, we use the float tensor and not integer tensor to represent the data else it would become a classification problem and not a regression problem as it should be.

6. ETA KPI
The number of output features correspond to the target size 281 * 191.

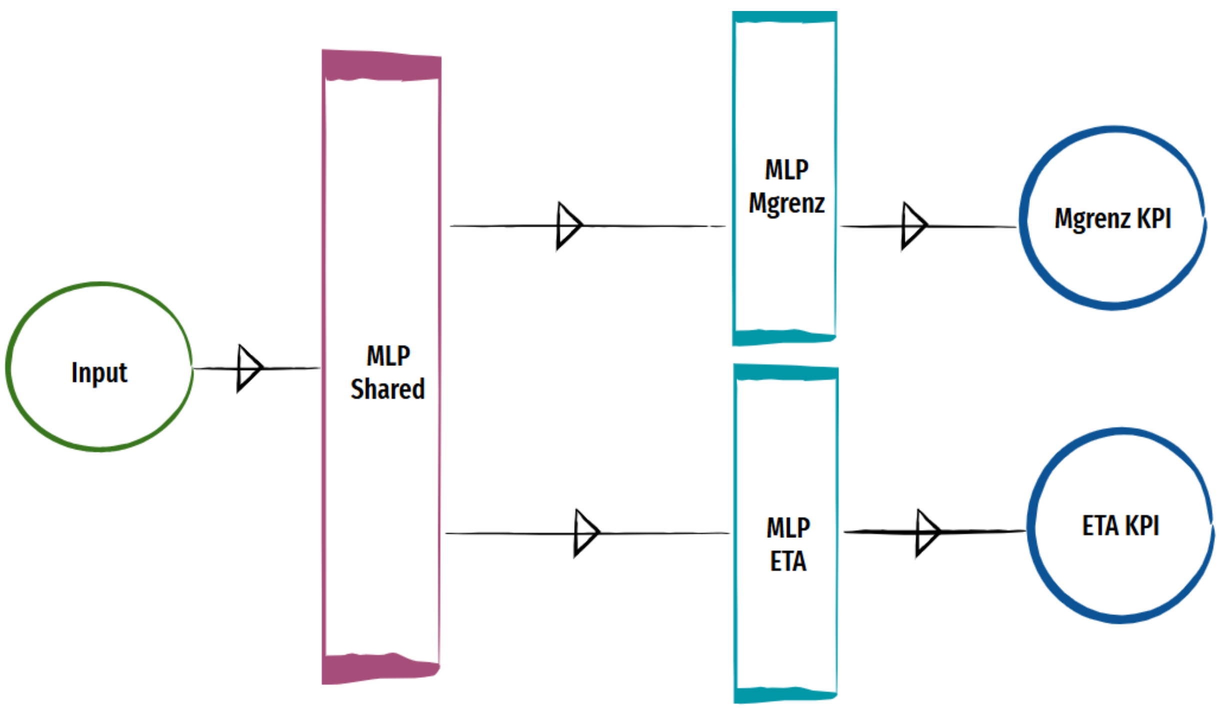

Figure 5.1: MLP Model Architecture

5.2 Loss Functions

The Mean Squared Error (MSE) loss is the loss function used for our problem with the intention that the squared losses penalize the model and inturn encourage it to minimize it further. In addition to its contribution in exaggerating the loss, MSE also ensures that deviations are positive and donot confuse the model by negating the losses of different signs.

5.2.1 Loss for Mgrenz KPI(Torque curve)

The MSE loss for the Mgrenz KPI is formulated as below :

$$\text{Y1 Loss} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h}\sum_{j=1}^{h}(y_{ij} - \hat{y}_{ij})^2 \tag{5.1}$$

where

- n: EM samples
- h: dimensionality of 1D vector

To encourage the model to learn the nature of the curve, we have experimented with 2 Loss Regularization techniques. Both the Regularizations are L2 Regularization to be in syn with the dynamics of MSE.

1. Smoothening Loss Regularization

   To smoothen out the curve for the Mgrenz KPI we apply a loss regularisation factor to take into account.

   $$\text{Y1 Smoothening Loss Regularization} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h-1}\sum_{j=1}^{h-1}\left(\text{ReLU}(|\hat{y}_{ij+1} - \hat{y}_{ij}| - 1)\right)^2 \tag{5.2}$$

   This nature is factored in by penalising the loss by the magnitude if the neighbouring values in the prediction are not close to each other.

2. Decreasing Loss Regularization

   We used a different regularization approach to take into consideration the almost continuous decreasing nature of the curve.

   $$\text{Y1 Declining Loss Regularization} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{h-1}\sum_{j=1}^{h-1}\left(\text{ReLU}(\hat{y}_{ij+1} - \hat{y}_{ij})\right)^2 \tag{5.3}$$

   The Torque curve closely resembels a decreasing sigmoidal curve and hence we use this knowledge to penalize the loss for non-decreasing values within each prediction.
   CITE!!!! If the electromagnetic coil is enabled by the commutator for the time span t3, the (almost) maximal current is running through it's loops and the (almost) maximal magnetic field strength is generated. The (almost) maximal torque is acting on the rotor. If the time span is shortened to t2 by increasing rotational speed, a slightly lower torque is acting, because the current through the coil is decreasing slightly. When reducing the time span to t1, the coil gets disconnected from the input voltage even though just half the maximum current is reached. Accordingly the torque decreases significantly:

We have not combined both the above regularizations as they donot complement each other. This is because the loss regularized by 5.2 will not necessarily be a decreasing curve. This holds true for the regularization 5.3 as it may not necessarily have gradual transitions in the curve. Nevertheless, we perform ablation studies with both the regularizations and report the results obtained in 6.3

5.2.2 Loss for ETA KPI(Efficiency Grid)

The MSE loss for the 3 Dimension (3D) KPI is formulated as below :

$$\text{Y2 Loss} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{w}\frac{1}{h}\sum_{j=1}^{w}\sum_{k=1}^{h}(M_{ijk}\cdot y_{ijk}) - (M_{ijk}\cdot \hat{y}_{ijk}))^2 \tag{5.4}$$

$$M_{ijk} = \begin{cases} 1 & \text{if } y_{ijk} \neq \text{NaN} \\ 0 & \text{if } y_{ijk} = -1 \end{cases} \tag{5.5}$$

where

- M_{ijk} *: Mask matrix*
- w *: 1st dimension of 3D vector*
- h *: 2nd dimension of 3D vector*

The ETA KPI is a 3D plot of real numbers ranging between 0 and 100.NEED TO CITE...EVEN IF IT IS A KNOWN FACT.
We noticed in some portions of the ETA KPI, the plot not visible as it had NaN values.
As ANN cannot be trained to predict NaN values we have a binary mask constructed such that values corresponding to NaN in the target have value 0 and all other values as 1. Mathematically, this process can be expressed as is in Equation 5.5 and thus ensure that the NaN values are ignored in the loss calculation. The mask is then multiplied with both the target and its respective prediction.

Additionally, to encourage the model to learn the nature of the ETA KPI from our observations gathered in 3.1.3, we have tried to incorporate all of the below learnings via the loss function as Y2 Regularization.

1. Efficiency at Maximum Torque Loss Regularization

   To ensure that the shape of the ETA KPI is maintained, we also regularize the loss for the maximum torque value. To do so, we have attempted to retrieve the last rows our ETA KPI and those of its target values and penalise the squared difference to have higher weight. We formulate it mathematically as below:

$$\text{Y2 Loss Regularization MM Max Mgrenz} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{t1}\sum_{j=-t1}^{w}\frac{1}{h}\sum_{k=1}^{h}(y_{ijk} - \hat{y}_{ijk})^2 \tag{5.6}$$

   where

   - $t1$ *: Threshold for inital ETA KPI envelope boundary*

   The number of last rows is determined by a threshold t1

2. Efficiency along the shape of the ETA KPI envelope

   To ensure that the envelope follows the shape of the Mgrenz KPI curve, we penalise the model heavily in case of deviations at the border. This was developed by considering the last columns for each row of the ETA grid and penalising the squared difference with that of the target.

   We formulate it mathematically as below:

$$\text{Y2 Loss Regularization Envelope} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{w}\sum_{j=1}^{w}\frac{1}{t2}\sum_{k=-t2}^{h}(y_{ijk} - \hat{y}_{ijk})^2 \tag{5.7}$$

where

- *t2 : Threshold for ETA KPI envelope*

The number of last rows is determined by a threshold t2

3. Efficiency at low speeds

To force the model to pay more attention at lower speeds, we have regularized the loss for the first few columns of each row of the ETA grid and penalise the squared difference with that of the target.

We formulate it mathematically as below:

$$\text{Y2 Loss Regularization Low Speed} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{w}\sum_{j=1}^{w}\frac{1}{t3}\sum_{k=1}^{t3}(y_{ijk}-\hat{y}_{ijk})^2 \tag{5.8}$$

where

- *t3 : Threshold for Low Speed*

The number of first columns is determined by a threshold t3.
It is a known fact that at 0 Torque, the corresponding efficiency values for the motor is 0. With this regularization, this learning as well is incorporated into the loss function.

4. Efficiency at low Torque

To force the model to be more careful at low torque, we have regularized the loss for the first few rows of each column of the ETA KPI and penalise the squared difference with that of the target.

We formulate it mathematically as below:

$$\text{Y2 Loss Regularization Low Torque} = \frac{1}{n}\sum_{i=1}^{n}\frac{1}{t4}\sum_{j=1}^{t4}\frac{1}{h}\sum_{k=1}^{h}(y_{ijk}-\hat{y}_{ijk})^2 \tag{5.9}$$

where

- *t4 : Threshold for High Torque*

The number of first rows is determined by a threshold t4

The above Y2 Regularizations are indeed purely MSE but with higher weights for specific regions of the ETA KPI.
Consequently being L2 Regularizations it goes hand in hand with MSE Loss calculated in 5.4.

$$\begin{aligned}\text{Y2 Loss Regularization} = {} & \text{Y2 Loss Regularization MM Max Mgrenz} + \text{Y2 Loss Regularization Envelope} + \\ & \text{Y2 Loss Regularization Low Speed} + \text{Y2 Loss Regularization Low Torque}\end{aligned} \tag{5.10}$$

$$\begin{aligned}\text{Total Loss} = {} & \text{w} \times (\text{Y1 Loss} + (\lambda_{\text{y1}} \times \text{Y1 Loss Regularization})) + \\ & (1 - \text{w}) \times (\text{Y2 Loss} + (\lambda_{\text{y2}} \times \text{Y2 Loss Regularization}))\end{aligned} \tag{5.11}$$

where

λ_{y1}: Y1 Loss Regularization Weight

λ_{y2}: Y2 Loss Regularization Weight

w: Y1 Loss Weightage

1 - w: Y2 Loss Weightage

We have added a Weightage parameter that controls the contribution of the Y1 Loss and Y2 Loss to the Total Loss. There are 2 reasons why this is useful for us :

1. When the targets are not of the same scale.
   However, without scaling, the losses are drastically different which we could not control with the weightage parameter and therefore after much pondering we decided to scale the targets.
2. When the prediction accuracy of one KPI is sunstantially more important than the other.
   Our task demands the same as the ETA KPI is post processed to be within the shape of the Mgrenz KPI. Therefore, ideally have higher weightage for the Mgrenz KPI as its loss in performing well is costlier.

Furthermore, the weightage parameters for both target is designed to sum upto 1 keeping in mind improved training stability as a result of normalized weights.

Finally the aggregated loss is backpropagated.

5.3 Optimizer

Adam optimizer is used for optimization as it is known to be computationally efficient and requires little memory. [17]
It infers the gradients of the loss and how it impacts the weights and biases of each layer and thus guide the model to decrease the loss.
The optimizer acts once the loss is backpropagated across training each batch of the dataset.
We have also experimented with an exponential learning rate scheduler which reduces the learning rate exponentially by a gamma parameter to decay learning as training progresses across epochs.

5.4 Evaluation Metrics

The evaluation metrics we have considered for our regression problem is the RMSE.
Therefore, the model with the least prediction scores ie, closest to 0 is ideal for our application.

5.4.1 Evaluation Metrics for Mgrenz KPI

The Y1 Score for the Mgrenz KPI is formulated as below :

$$\text{Y1 score} = \frac{1}{n}\sum_{i=1}^{n}\underbrace{\sqrt{\frac{1}{h}\sum_{j=1}^{h}(y_{ij}-\hat{y}_{ij})^2}}_{y1\ rmse} \tag{5.12}$$

where

- *n : EM samples*
- *h : dimensionality of 1D vector*
- *y1 rmse: RMSE for each test sample*

5.4.2 Evaluation Metrics for ETA KPI

The Y2 score for the ETA KPI is formulated as below :

$$\text{Y2 score} = \frac{1}{n}\sum_{i=1}^{n} \underbrace{\sqrt{\frac{1}{w}\frac{1}{h}\sum_{j=1}^{w}\sum_{k=1}^{h}(y_{ijk} - \hat{y}_{ijk})^2}}_{y2\ rmse} \tag{5.13}$$

where

- *n : EM samples*
- *w : 1st dimension of 2D vector*
- *h : 2nd dimension of 2D vector*
- *y2 rmse : RMSE for each test sample*

5.5 Post Processing

The mean and standard deviation from the train-validation datasets are applied to transform the test dataset to maintain uniformity in the predictions generated. In the case of new files we first convert it into the tabular representation our model consumes and then apply the scaling.
Hence the reason why we preserve the same scalers used during training as we not only evaluate our dedicated test dataset but also for clients to use on demand.

Furthermore as part of post processing, we slice the predicted ETA KPI to only contain the values within its predicted Mgernz KPI. Since we are predicting a padded matrix to ensure dimensionality sync across different ETA KPI's, the grid contains values even outside the boundary of the ETA KPI. Hence, we attempted to slice the shape of the Mgrenz KPI curve from the ETA KPI by counting the number of columns a row to have based on consecutive values in the curve. This brings us back to the point that it is imperative the prediction of the Mgrenz KPI is close to perfect as the envelope of the ETA KPI inherently is dependent on it.

Chapter 6

Experiments and Results

6.1 Experiments with MLP

The learnable parameters were chosen via a random grid search and was tuned by monitoring the model's performance across 5 fold cross validation training.
The splits are saved locally and can be used later to ensure reproducibility.

Hyperparameters	**Description**	**Value**
lr	Learning Rate	0.00075
hidden_sizes	Dimensionality of Hidden Layers	256
lr_gamma	Exponential Learning Rate Scheduler	0.9
batch_size	Batch Size	72
epochs	Number of Epochs	8
p	Dropout Probability	0.2
lambda_y1	Y1 Loss Regularizer	6
lambda_y2	Y2 Loss Regularizer	3
w_y1	Weightage of Y1 Loss	0.8

Table 6.1: Hyperparameter Tuning

We chose `Wandb`[1] to log metrics from the training run and to monitor model performance across folds. Below is the visualisation of the training and validation metrics for both KPIs.

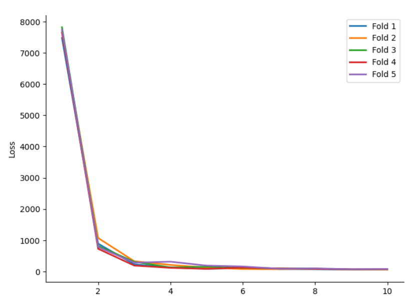

Figure 6.1: Aggregated Training Loss

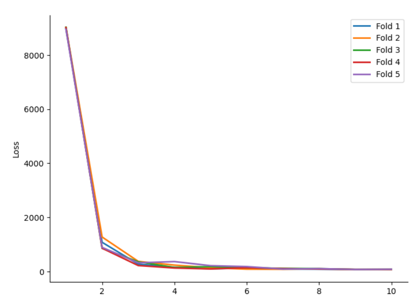

Figure 6.2: Training Loss for Mgrenz KPI

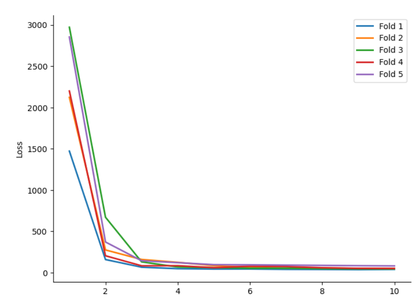

Figure 6.3: Training Loss for ETA KPI

[1]Weights & Biases

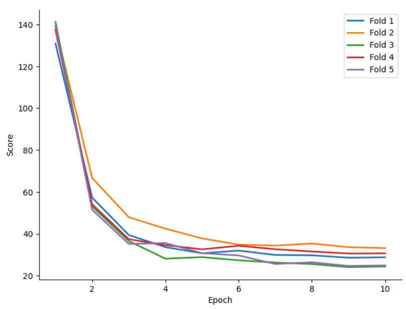

Figure 6.4: Aggregated Training Score

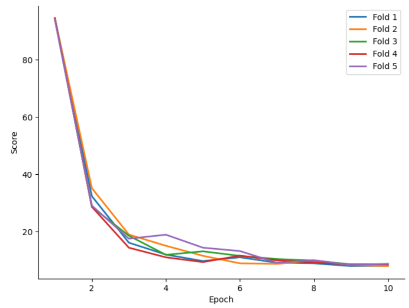

Figure 6.5: Training Score for Mgrenz KPI

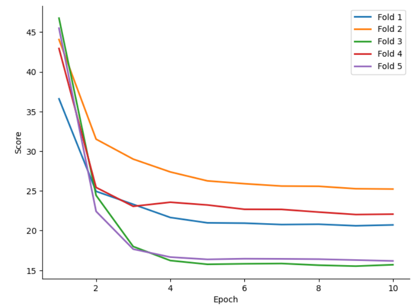

Figure 6.6: Training Score for ETA KPI

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

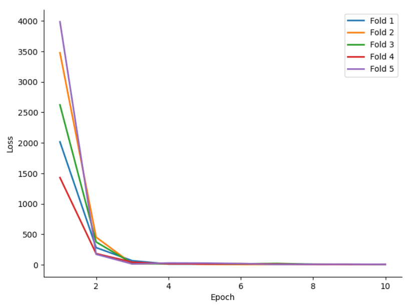

Figure 6.7: Aggregated Validation Loss

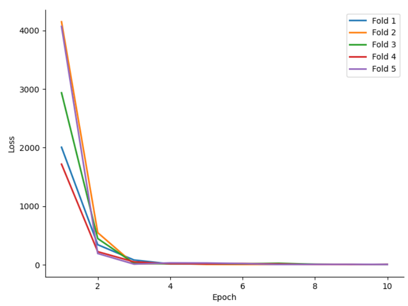

Figure 6.8: Validation Loss for Mgrenz KPI

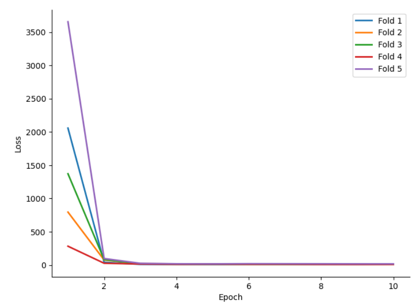

Figure 6.9: Validation Loss for ETA KPI

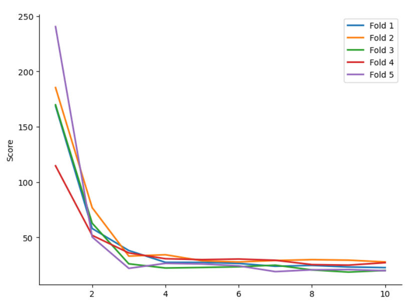

Figure 6.10: Aggregated Validation Score

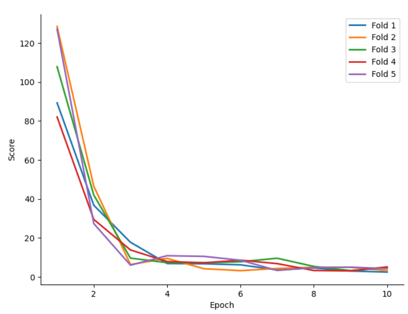

Figure 6.11: Validation Score for Mgrenz KPI

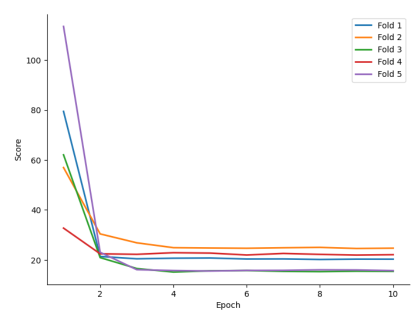

Figure 6.12: Validation Score for ETA KPI

We see a good fit of the model to the data with corresponding y1 and y2 scores approaching close to 0. We have also enabled saving the best performant model locally so it can be loaded on demand by the client when in need to only run inference.
We have narrowed down scoring to follow the criteria as depicted in Table 6.2.

Percentage Difference	0-5%	5-10%	10-15%	15-20%	20-25%	25-30%	30-35%	35-40%	40-100%
Y1 Score	0-11	11-22	22-33	33-44	44-55	55-66	66-77	77-88	>88
Y2 Score	0-5	5-10	10-15	15-20	20-25	25-30	30-35	35-40	>40

Table 6.2: Scoring Criteria

This is deduced from the below equation

$$\text{Percentage Difference} = (\text{Score}/(Max - Min)) \times 100 \tag{6.1}$$

From our observations the values for y1 targets range between 50 - 280 and those of the y2 target range between 0 - 100.

6.2 Results with MLP

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

6.2.1 Mgrenz KPI Results with MLP

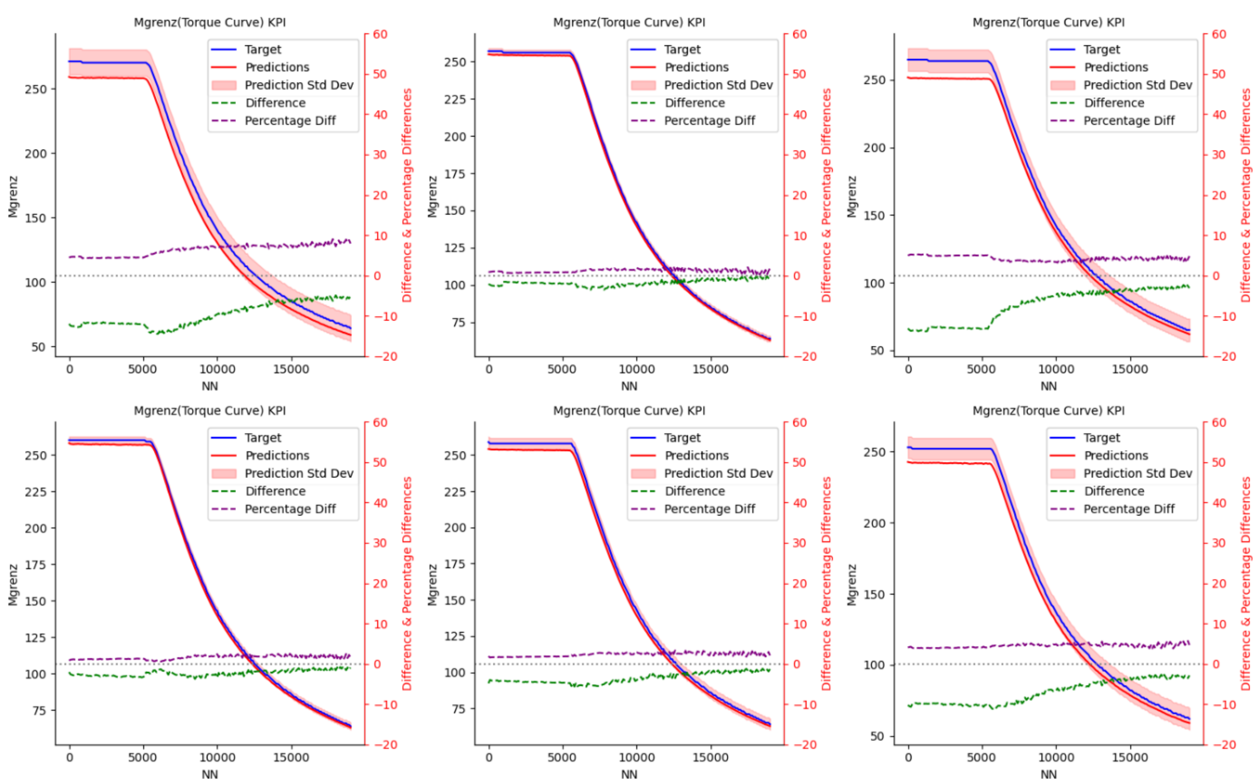

Figure 6.13: MLP Training Results for 2D KPI(Mgrenz)

The Average RMSE and element wise RMSE for the test dataset performance with the MLP is as below:

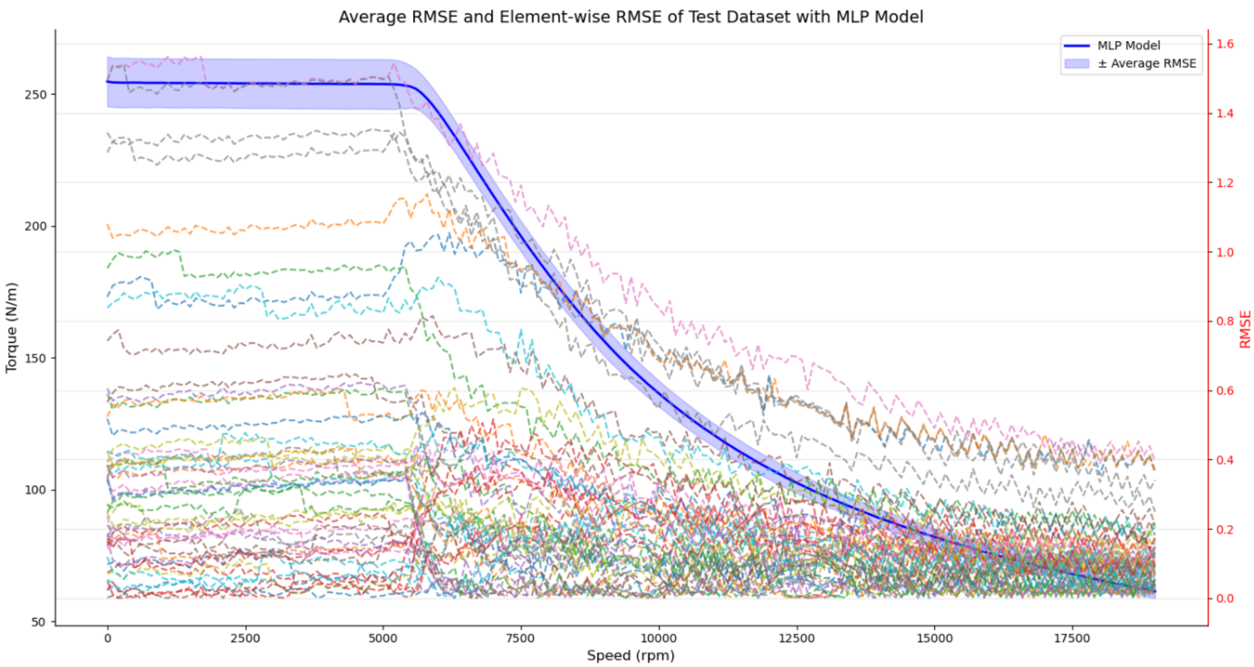

Figure 6.14: Test Dataset RMSE Evaluation for 2D KPI(Mgrenz)

Figure 6.15 shows the score statistics of the model performance of Mgrenz KPI over the test dataset.

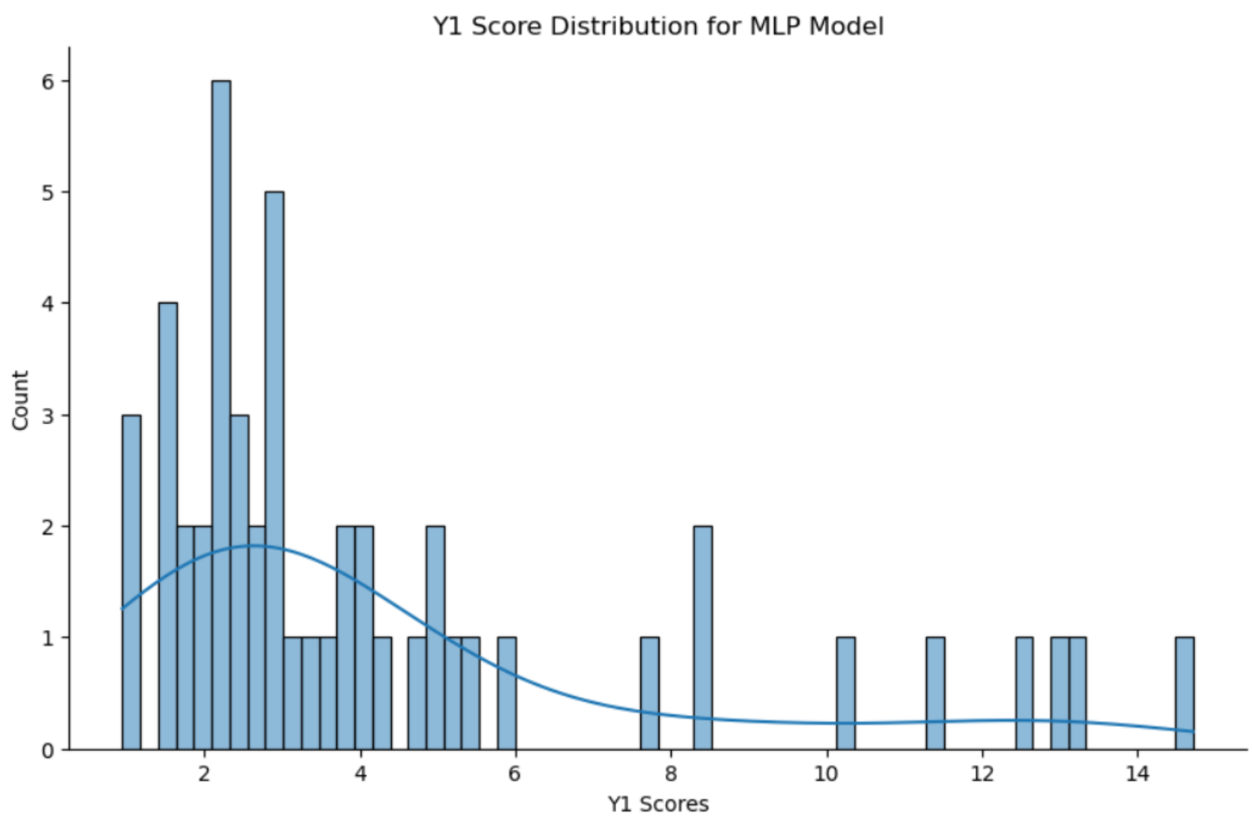

Figure 6.15: MLP Score statistics for 2D KPI(Mgrenz)

6.2.2 ETA KPI Results with MLP

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

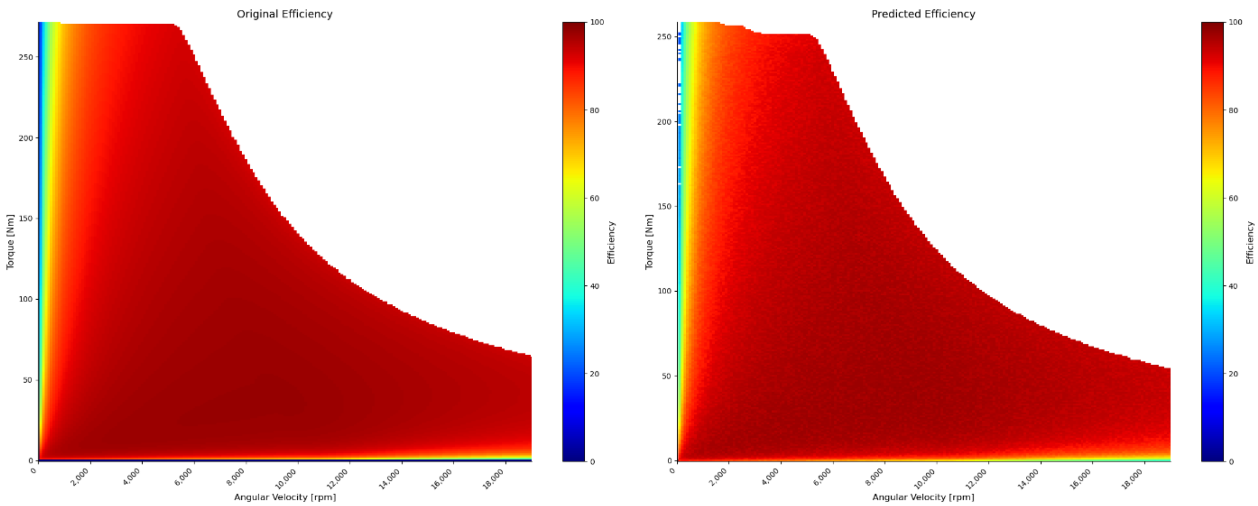

Figure 6.16: 1st MLP Training Results for ETA KPI

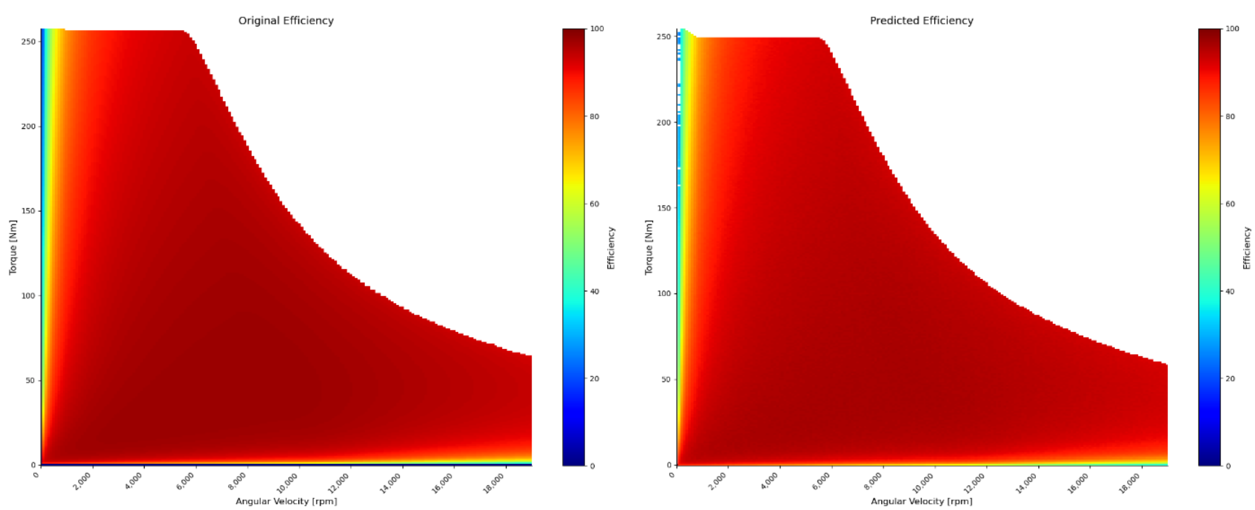

Figure 6.17: 2nd MLP Training Results for ETA KPI

Below figures shows the overlap difference of the target with the prediction.

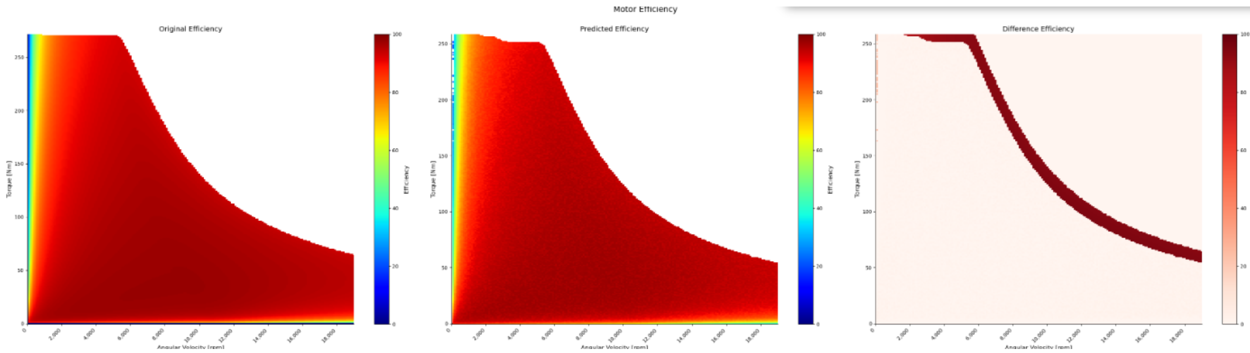

Figure 6.18: 1st Eval MLP Training Results for ETA KPI

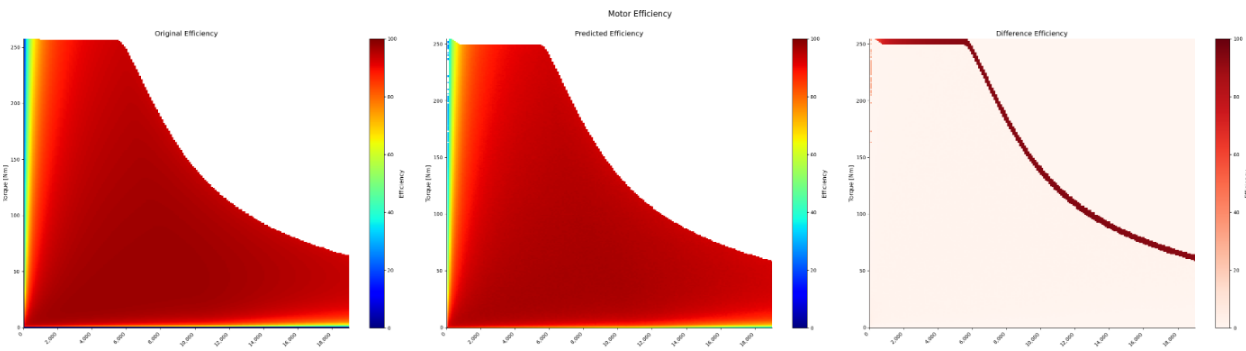

Figure 6.19: 2nd Eval MLP Training Results for ETA KPI

ETA KPI are harder to evaluate scoring as it is a 3D plot. Therefore, to visualize the efficiency prediction deviation with its respective targets we do so for specific speeds across the entire torque range eith Figure 6.20

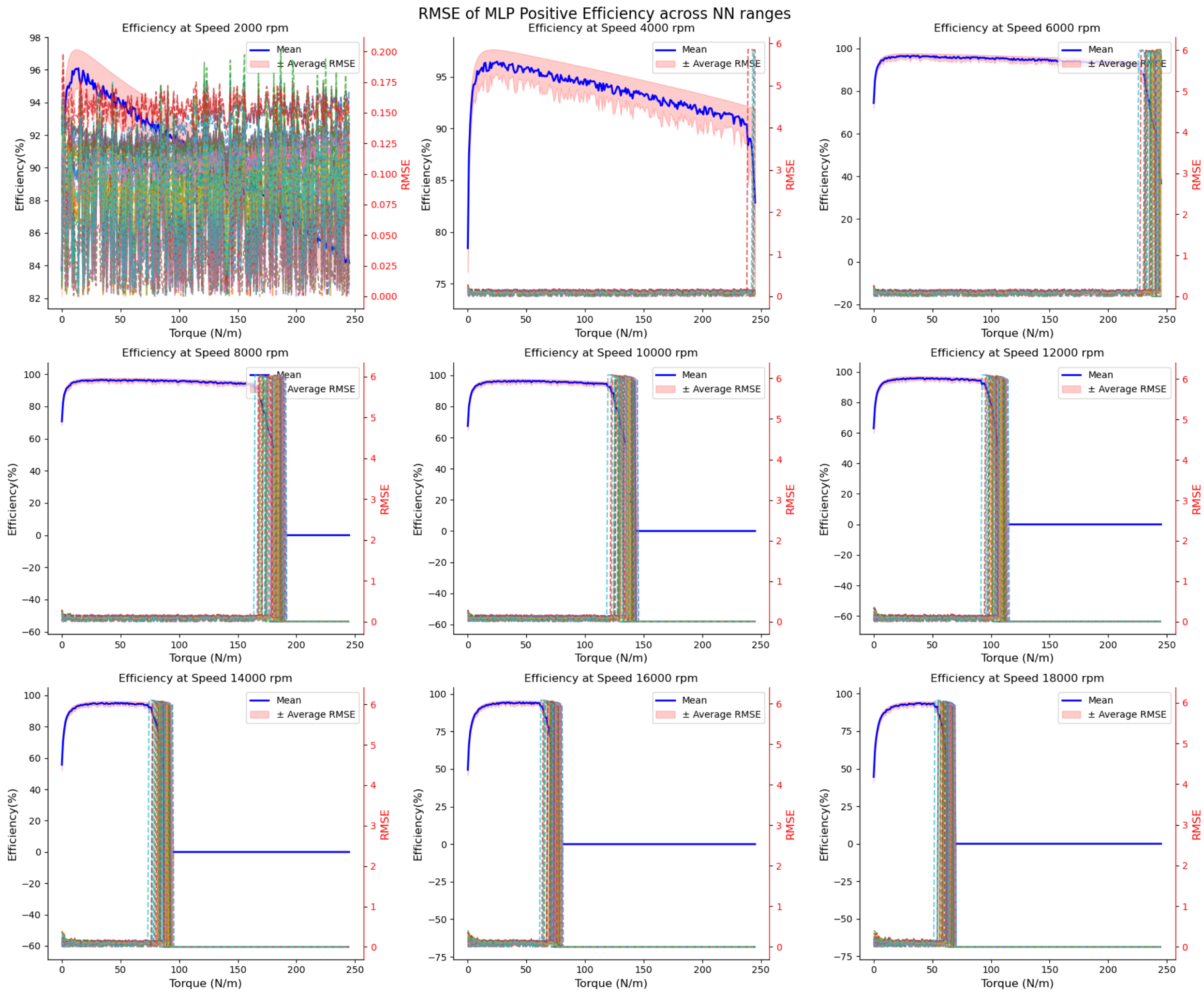

Figure 6.20: Eval MLP ETA RMSE KPI

Observations from the predictions helped to correct few discrepancies in our development for instance in the ETA grid we replaced 0 with NaN values which we later understood were both represented different in the grid.
As Efficiency values can take up values only between 0 and 100, we consider the same as constant across plots and use it as a baseline for determining the levels in the contour plot.
We have also left the output predictions for the Torque curve to remain as float values even when the target values are integers to preserve data precision. We give the client the flexibility to turn this on/off demand.
Figure 6.21 shows the score statistics of the model performance of ETA KPI over the test dataset.

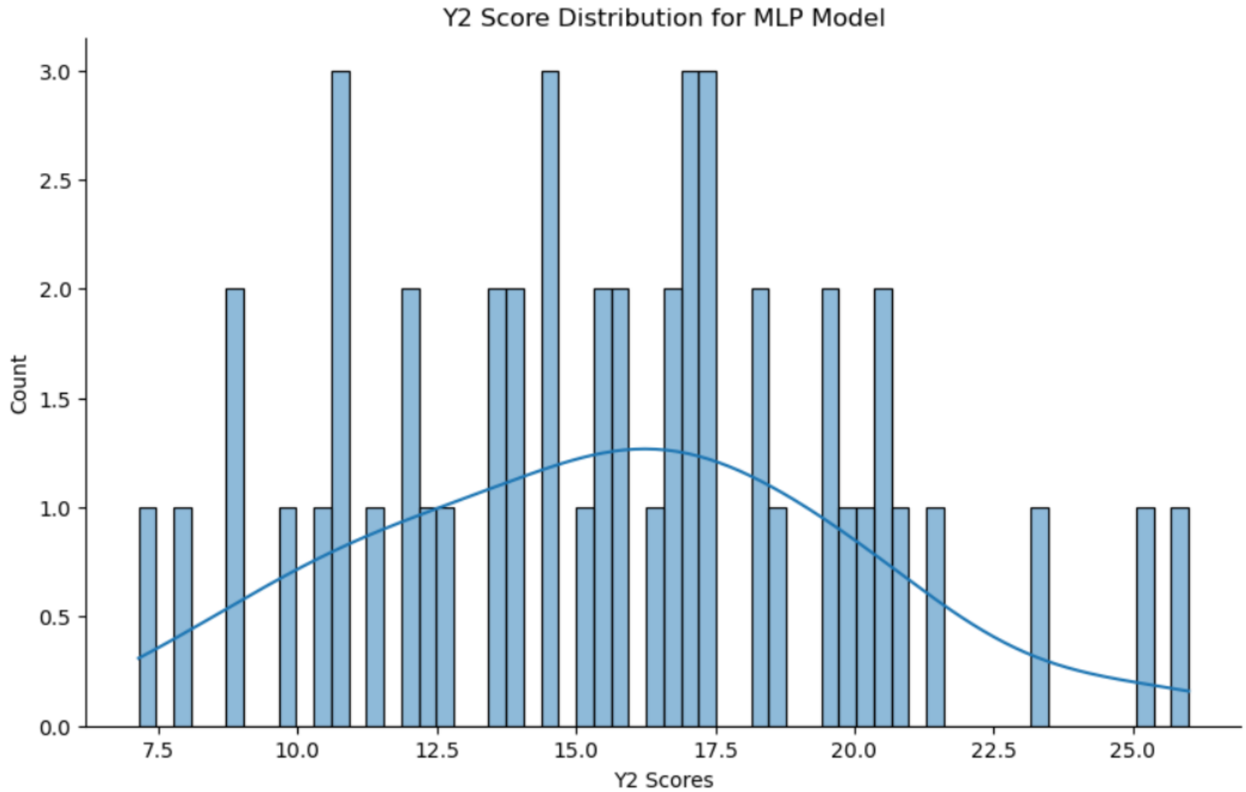

Figure 6.21: MLP Score statistics for ETA KPI

We have used python 3.12.2 for our development and the pytorch library compatible with cuda. The model was trained on a NVIDIA V100 GPU with blah blah.

6.3 Results with Baseline

From our observations of how the predictions closely resembled that of the target values, we have proceeded with a baseline model which is essentially the average of the train dataset.

$$\text{Y1 Element wise RMSE} = \sqrt{\frac{1}{n}(\bar{x} - x_i)^2} \quad \forall i \in \{0, \ldots, w-1\} \tag{6.2}$$

where

- n : *EM samples*
- w : *1st dimension of 1D vector*
- $\bar{x}$: *Baseline Average mean*

We also have the score statistics of how the Baseline model performs over the test dataset.

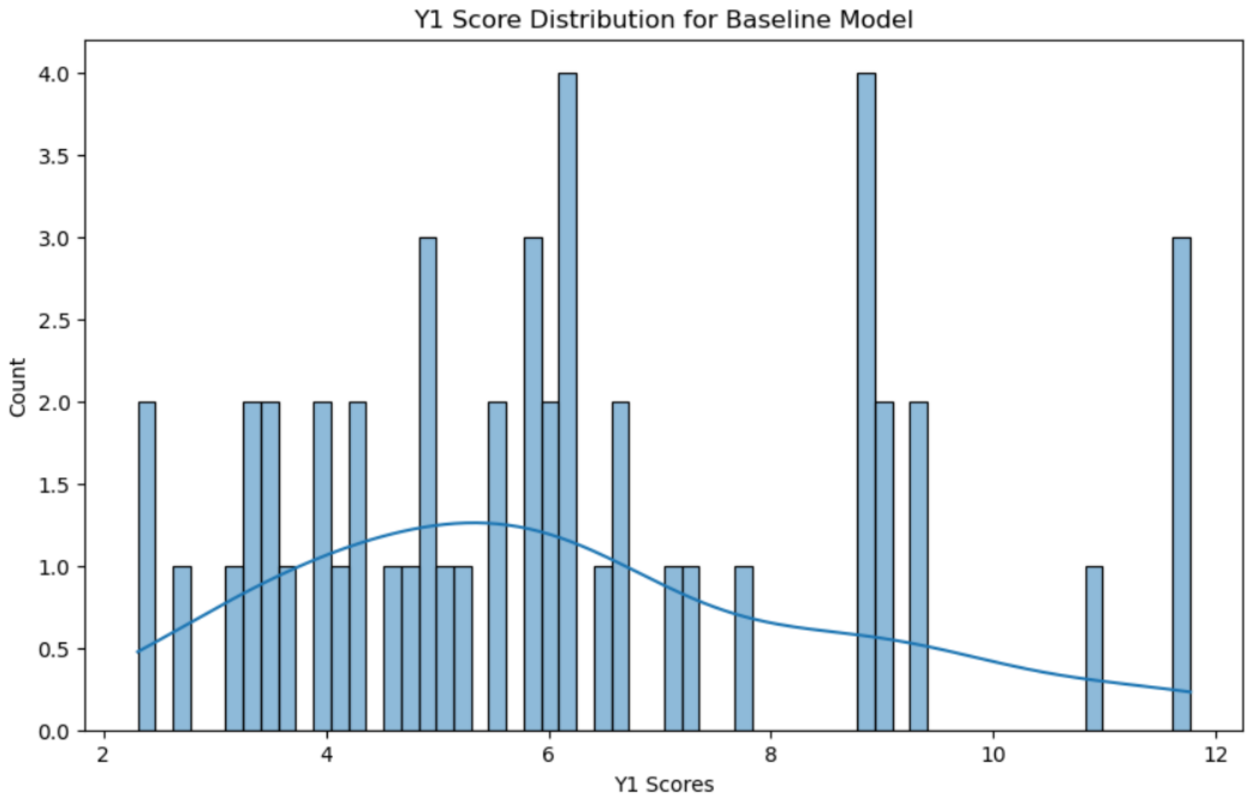

Figure 6.22: Baseline Score statistics for Mgrenz KPI

6.3.1 ETA KPI Results with Baseline

We visualize the efficiency baseline deviation with its respective targets for specific speeds across the entire torque range eith Figure 6.23

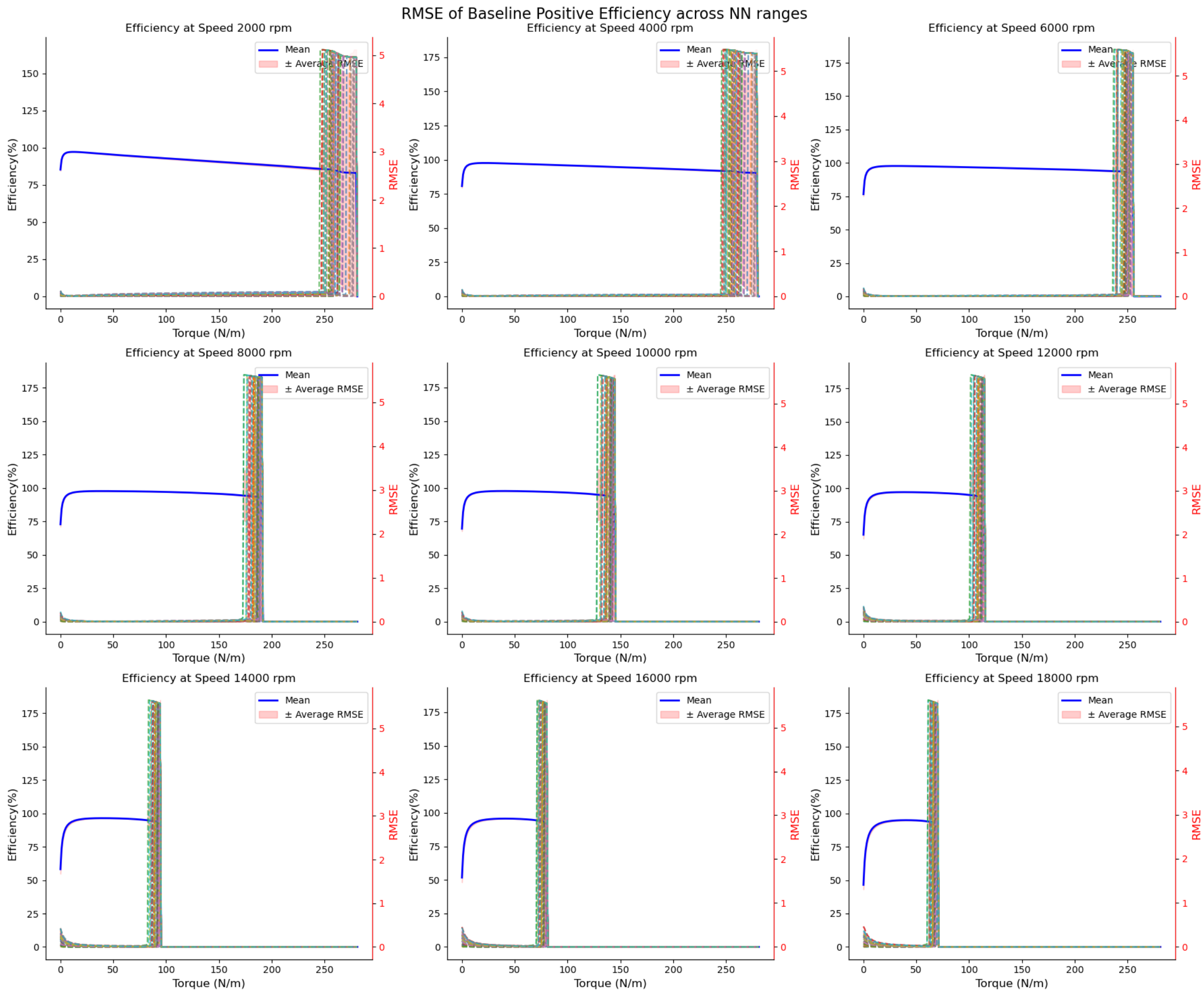

Figure 6.23: Eval Baseline ETA RMSE KPI

Figure 6.21 shows the score statistics of the model performance of ETA KPI over the test dataset.

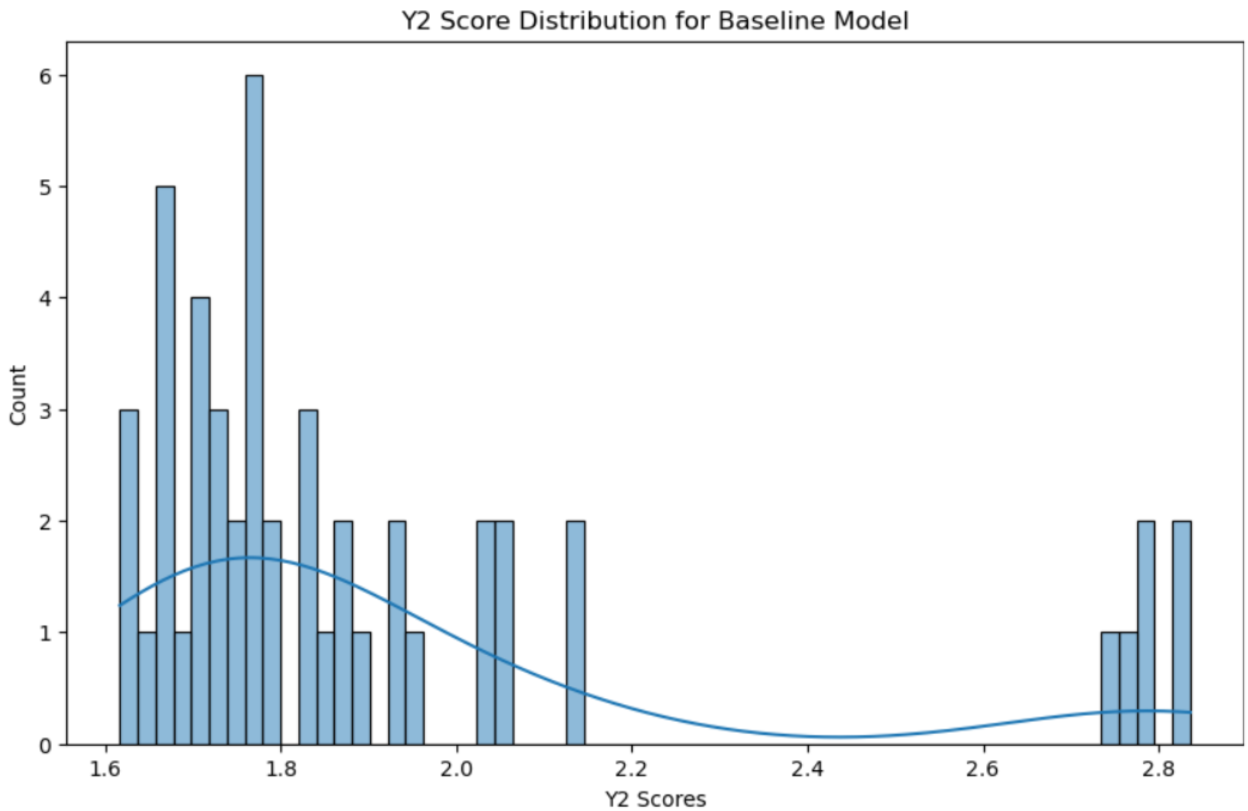

Figure 6.24: Baseline Score statistics ETA KPI

6.4 Ablation Studies

As part of ablation studies, we have compared our evaluations of both the MLP and the Baseline model on both targets respectively.

Model	Y1 Score	Y2 Score
Baseline	4.3656	4.3149
MLP	1.6886	3.1482
MLP[a]	2.7681	5.1206

Table 6.3: Ablation Studies

[a]MLP without Loss Regularization

We draw the inferences that our model has performed relatively better than the baseline for 2D KPI. The inferences shown are strictly speaking from the Double V Magnet Topology which assumed the bulk of the data we had received.

Chapter 7

Conclusion

This thesis offers a fresh outlook to the possibility of modelling the performance of an electric motor using GNNs. It also lays the foundation for future work on being able to generate electric motor design parameters conditioned on the 2 KPIs we predicted.

7.1 Future Improvements

I would suggest the following improvements to our study :

1. Build a model that uses the Mgrenz KPI prediction to predict its corresponding ETA KPI.
2. Although we have designed the model to be topology invariant for 3 topologies, we only had sufficient data from Double V Topology to draw evaluations from. Further evaluations on the other topologies would be beneficial to critically assess our model's performance.
3. Another interesting study would be to benchmark model performance when one backpropagates the losses for each KPI individually rather than scaling them.
4. Additionally, the motivation of building a topology invariant model was the reason we have considered building a heterogeneous graph to model the data. The machinery is elaborated in Section CIE. Such a model could serve as yet another ablation study to our problem.

Appendix

7.2 Data Preprocessing for GNN

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:
   $$sw = \{s_i w_j\} \quad \forall i \in \{1, \ldots, QSim\}, \quad \forall j \in \{1, \ldots, N\}$$
   - Slots:
   $$s = \{s_i\} \quad \forall i \in \{1, \ldots, QSim\}$$
   where
     - *Qsim : Count of slots in the Stator*
     - *N : Count of copper windings per slot*

3. **Rotor**

   - Magnet Flux Barriers:
   $$v = \{v_{ij}\} \quad \forall i \in \{1, \ldots, T\}, \quad \forall j \in \{1, \ldots, V\}$$
   - Magnets:
   $$vm = \{v_i m_j\} \quad \forall i \in \{1, \ldots, T\}, \quad \forall j \in \{1, \ldots, V\}$$
   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

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

   angle=vm-vm

2. **Distance**
   Relevant Paths

$$vi--vi = \{v_{ij_1} - v_{ij_2}\}, \forall i \in \{1,\dots,T\}, \forall j_1, j_2 \in \{1,\dots,V\} \mid j_1 \neq j_2$$
$$vi--vj = \{v_{i_1j} - v_{i_2j}\}, \forall i_1, i_2 \in \{1,\dots,T\}, \forall j \in \{1,\dots,V\} \mid i_1 \neq i_2$$
$$v--vm = \{v_{ij} - v_i m_j\}\forall i \in \{1,\dots,T\}, \quad \forall j \in \{1,\dots,V\}$$
$$v--rr = \{v_{ij} - r_r\}, \forall i, j \in \{1,\dots,T\}$$
$$o--r = \{(o - r_r), (o - r_a)\}$$
$$rr--s = \{r_r - s_i\}, \forall i \in \{1,\dots,QSim\}$$
$$s--sw = \{s_i - s_i w_j\}, \forall i \in \{1,\dots,QSim\}, \forall j \in \{1,\dots,N\}$$
$$s--ra = \{s_i - r_a\}, \forall i \in \{1,\dots,QSim\}$$
$$sw--sw = \{s_i w_{j_1} - s_i w_{j_2}\}, \forall i \in \{1,\dots,QSim\}, \forall j \in \{1,\dots,N\} \mid (j_1 == j_2 - 1)$$

   distance = vi–vi + vi–vj + v–vm + v–rr + o–r + rr–s + s–sw + s–ra + sw–sw

Node Features

1. **v** = {lmsov, lth1v, lth2v, r1v, r11v, r2v, r3v, r4v, rmt1v, rmt4v, rlt1v, rlt4v, hav}
2. **vm** = {mbv, mhv, rmagv}
3. **r** = {r}
4. **s** = {b_nng, b_nzk, b_s, h_n, h_s, r_sn, r_zk, r_ng, h_zk}
5. **sw** = {bhp, hhp, rhp}

Path Features

1. **vm–vm** = {deg_phi}
2. **vi–vi** = {dsm, dsmu}
3. **vi–vj** = {amtrvj-amtrvi}
4. **v–vm** = {lmav, lmiv, lmov, lmuv}
5. **v–r** = {amtrv, dsrv}
6. **o–r** = {r}
7. **rr–s** = {airgap}
8. **s–sw** = {dhphp}
9. **sw–sw** = {dhpng}
10. **s–ra** = {r_a-(r_i + h_n + h_zk)}

The heterogeneous graph that was constructed earlier is as below:

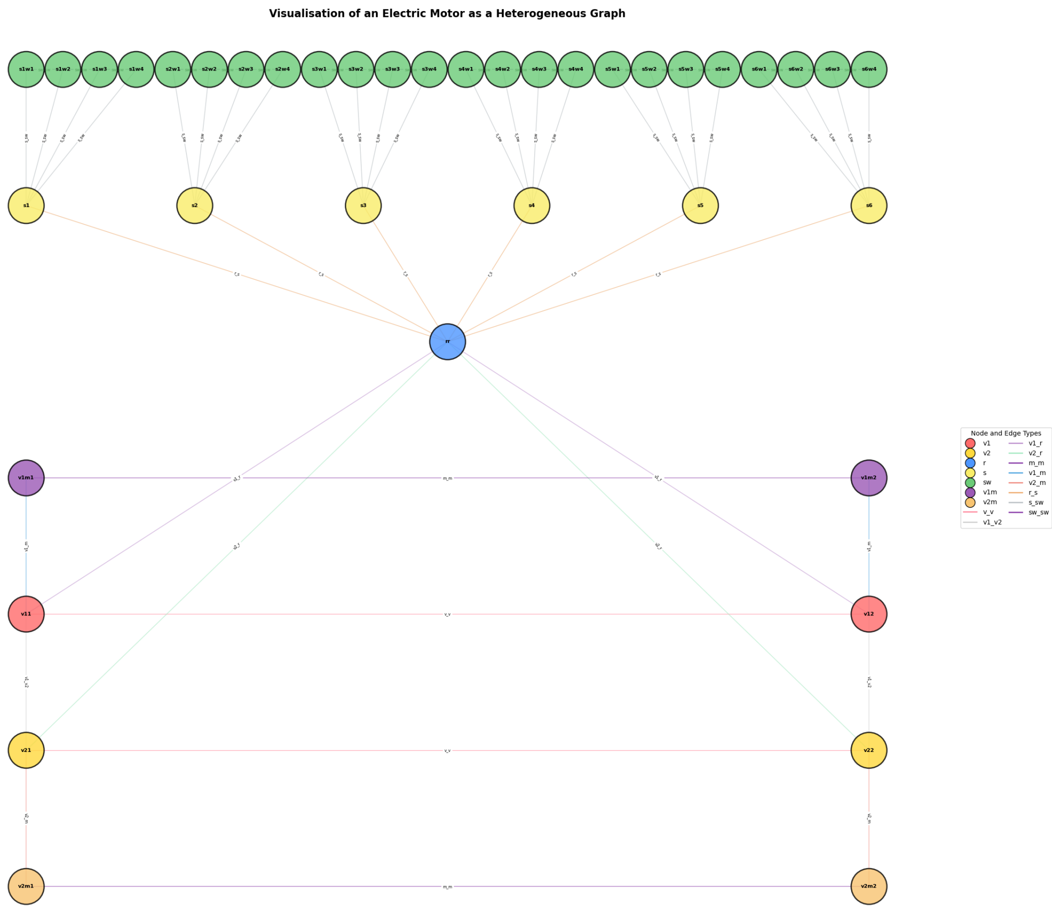

Figure 7.1: HetGraph

List of Figures

List of Tables

Bibliography

[01] Tsai Hor Chana, Guosheng Yina, Kyongtae Baeb, Lequan Yu *Multi-task Heterogeneous Graph Learning on Electronic Health Records.* cs.LG, 14 Aug 2024.

[02] Xiaocheng Yang, Mingyu Yan, Shirui Pan, Xiaochun Ye, Dongrui Fan *Simple and Efficient Heterogeneous Graph Neural Network.* cs.LG, 1 Sep 2023.

[03] Xinru Huang *A Heterogeneous Graph Neural Network Model for Electric Vehicle Purchase Propensity Prediction.* ICDSCA, Oct 2023.

[04] Qingsong Lv, Ming Ding, Qiang Liu, Yuxiang Chen, Wenzheng Feng, Siming He, Chang Zhou, Jianguo Jiang, Yuxiao Dong, Jie Tang *Are we really making much progress? Revisiting, benchmarking, and refining heterogeneous graph neural networks.* cs.LG, 30 Dec 2021.

[05] Fredrik Johannessen; Martin Jullum *Finding Money Launderers Using Heterogeneous Graph Neural Networks.* ICDSCA, 25 July 2023.

[06] Damian Owerko, Fernando Gama, Alejandro Ribeiro *Predicting Power Outages Using Graph Neural Networks.* cs.LG, Nov 2018.

[07] Mikko Tahkola, Janne Keränen, Denis Sedov, Mehrnaz Farzam Far, Juha Kortelainen *Surrogate Modeling of Electrical Machine Torque Using Artificial Neural Networks.* ICDSCA, Dec 17 2020.

[08] AKM Khaled Ahsan Talukder, Bingnan Wang and Yusuke Sakamoto *Electric Machine Two-dimensional Flux Map Prediction with Ensemble Learning.* ICEMS, 2022.

[09] Kazuhisa Iwata, Hidenori Sasaki, Hajime Igarashi, Daisuke Nakagawa, and Tomoya Ueda *Generalization Performance in Predicting Torque Characteristics Using Convolutional Neural Network and Stator Magnetic Flux.* ICDSCA, 3 Mar 2024.

[10] Simone Ferrari, Paolo Ragazzo, Gaetano Dilevrano, Gianmario Pellegrino *Flux-Map Based FEA Evaluation of Synchronous Machine Efficiency Maps.* WEMDCD, 2021.

[11] Vivek Parekh, Dominik Flore, Sebastian Schöps *Deep Learning-Based Prediction of Key Performance Indicators for Electrical Machines.* Jan 22, 2021.

[12] Carlos Candelo-Zuluaga, Antonio Garcia Espinosa, Jordi-Roger Riba, Pere Tubert Blanch *Computationally Efficient Analysis of Spatial and Temporal Harmonics Content of the Magnetic Flux Distribution in a PMSM for Efficiency Maps Computation.* 2020.

[13] Yihao Xu, Bingnan Wang, Yusuke Sakamoto, Tatsuya Yamamoto, and Yuki Nishimura *Comparison of Learning-based Surrogate Models for Electric Motors.* COMPUMAG, 2023.

[14] Marius Stender, Oliver Wallscheid, Joachim Böcker *Accurate Torque Estimation for Induction Motors by Utilizing a Hybrid Machine Learning Approach.* PEMC, 2021.

[15] Katsuyuki Narita, Hiroyuki Sano, Nicolas Schneider, Kazuki Semba, Koji Tani, Takashi Yamada, Ryosuke Akaki *An Accuracy Study of Finite Element Analysis- based Efficiency Map for Traction Interior Permanent Magnet Machines*: 2020.

[16] Jo Asanuma, Shuhei Doi, Hajime Igarashi *Transfer Learning Through Deep Learning: Application to Topology Optimization of Electric Motor*. 3 Mar 2020.

[17] Diederik P. Kingma, Jimmy Ba *Adam: A Method for Stochastic Optimization*. 30 Jan 2017.

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