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 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 Key Performance Indicators(KPIs). The KPIs are 2D and 3D plots on Torque(Mgrenz) and Efficiency(ETA). Other KPIs can be calculated from these two KPIs. For instance the Vibration Costs are inversely proportional to the Efficieny values predicted.

Abstrakt

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 Key Performance Indicators(KPIs). The KPIs are 2D and 3D plots on Torque(Mgrenz) and Efficiency(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 and Valeo for providing the data. Special thanks to Mr Daniel and Leo for sharing valuable insights of the data from a electromechanincal standpoint

Contents

Introduction

KPIs of an Electric Motor are essential to judge the performance of the motor before it is manufactured. Currently the KPIs are calculated by simulating the motor using an appropriate software. Although this works yet it is a very time consuming and expensive process as simulation is carried out in different iterations and optimized using a genetic algorithm.

Background

There has been extensive research in modeling the Electric Motor with Convolution Neural Networks(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 generating images is not ideal for our usecase. 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. 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. Although we see progress of GNNs in molecular chemistry, social networks usecases from which we draw inspiration from.

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

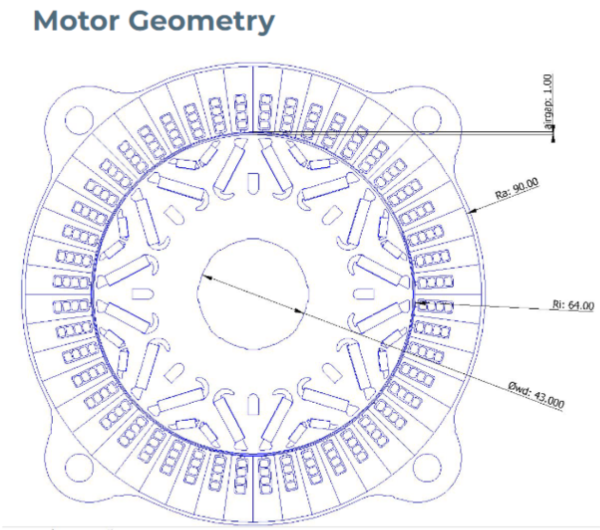

Figure 1: EM Geometry

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

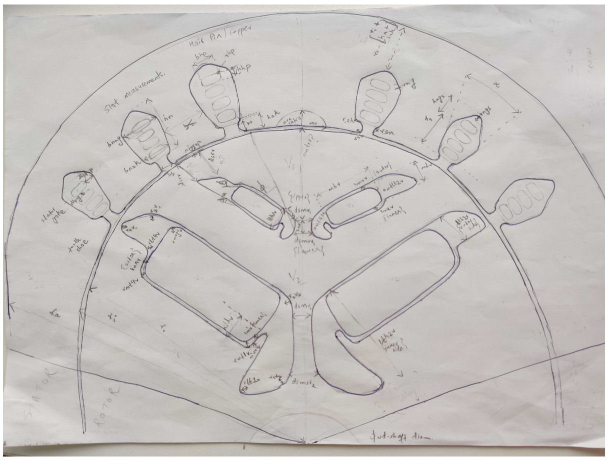

Figure 2: 1/8 Motor Crossection

For modelling the GNN, we represent the data in the form of 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:
     $$sw = \{s_i w_j\} \quad \forall i \in \{1, \ldots, QSim\}, \quad \forall j \in \{1, \ldots, N\}$$
   - 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**

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

   angle=vm-vm

2. **Distance**

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

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}

Edge 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:

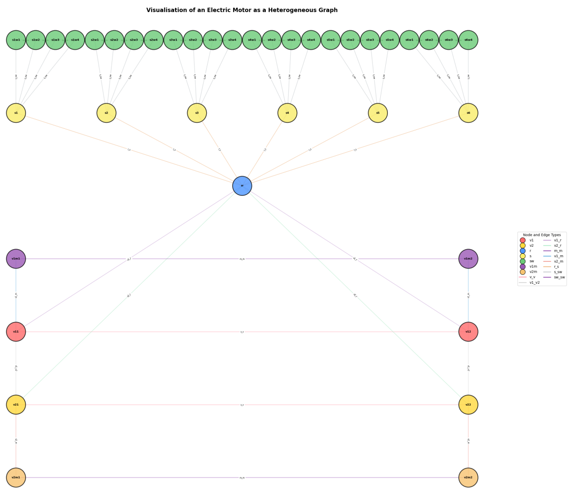

Figure 3: HetGraph

Modelling

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.

Experiments and Results

Conclusion

List of Figures

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