Temperature prediction of permanent magnet synchronous motor *

Yash Shah
Lakehead University
ID: 1161968

Dhruv Patel
Lakehead University
ID:1141257

Dhrumil Patel
Lakehead University
ID: 1133311

I. ABSTRACT

Most traction drive applications lack accurate temperature monitoring capabilities, requiring expensive large motor designs to ensure safe operation. Traditional thermal modelling necessitates knowledge of model parameter selection, which is influenced by factors such as motor geometry, cooling dynamics, and hot spot characterization. More- over, their major advantage over data-driven techniques, which is physical interpret ability, tends to diminish as soon as their degrees of freedom are reduced in order to satisfy the real-time need. In this article we have used KNN model, Polynomial Regression and Linear Regression model to find the best predictive analytic pipeline.

Keywords - Machine Learning, Deep learning, Temperature estimation, K-nearest Neighbour Regressor, Polynomial Regression, Linear Regression.

II. INTRODUCTION

In most automotive motor applications, high power and torque densities as well as high efficiency are required, which makes the permanent magnet synchronous motor (PMSM) the preferred choice.[1]Temperature-sensitive components, on the other hand, are vulnerable to failure under extreme thermal stress and must be closely monitored.Thermal sensors are used to monitor the temperature of the stator, but they are normally permanently placed in the stator, making replacement impossible, even though sensor functionality deteriorates over time.There is a financial motivation pushing the investigation of sufficiently accurate real-time temperature estimation because competitive pressure demands constant cost reduction.As a result, today's measure against excessive heat is a costly safety margin in embedded material. As a result, a PMSM's overload potential is never fully realised. This problem, combined with the automotive industry's ongoing cost pressure, pushes the development of sufficiently precise real-time temperature estimation technologies.

Direct thermal analyzes can be carried out using computational fluid dynamics (CFD) and heat equation inite element analysis (FEA)[2]. However, FEA and CFD demand a lot of computing power, making them unsuitable for real-time monitoring in the short and medium term.Lumped-parameter thermal networks are a good alternative for determining critical component temperatures (LPTNs). These are based on the idea of using comparable circuit diagrams to abstract the heat transfer mechanisms and simplify the complex thermal behaviour of a particular system. LPTNs can be gained in two ways: On the one hand white-box LPTN approaches are based on geometrical and material parameter information resulting in differential-algebraic models of high degrees [3].To calculate these under real-time limitations, significant computational resources are required, which often surpass the capabilities of industrial processing platforms. Grey-box LPTNs, on the other hand, contain empirical training data based on excessive motor measurements, resulting in significantly lower model orders but higher system abstraction.

Research attempts that vary from thermodynamic theory have also been made in recent decades: The setup of electric machine models that indirectly offer information about temperature-sensitive electrical model parameters is a typical lightweight method from this sector. There are ways that use current injection or voltage injection to acquire the stator winding resistance or the magnetization level of the magnets as thermal indicators, respectively, at the expense of additional losses.

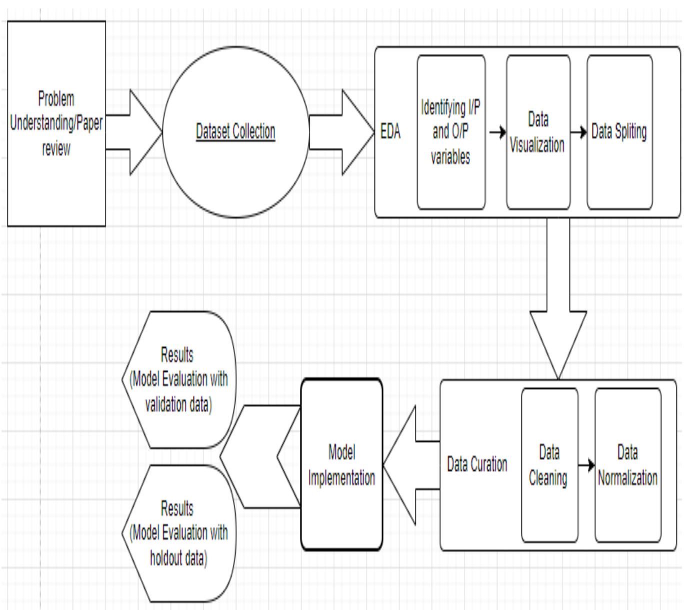
III. LITERATURE REVIEW

Certain ML approaches for the task of temperature profile estimation in a PMSM were studied before: Recurrent neural networks with memory units, in particular, long short-term memory (LSTM) or gated recurrent units (GRU) were evaluated on low-dynamic temperature profiles with a hyperparameter optimization via particle swarm optimization (PSO) in [4]. In [5], temporal convolutional neural networks (TCN) were applied on also high-dynamic data and a comparison with recurrent architectures were compiled after tuning hyperparameters with Bayesian optimization. Far simpler ML models like linear regression were also shown to be effective as long as data has been preprocessed with low-pass filters in [6].

This research builds on previous work by shedding light on the broader subject of supervised learning in regression challenges[4]. All previous papers have been classified as either fast and simple least-squares regression or computationally intensive and sophisticated deep learning, however there is a wide range of tools in between[6]. Models of intermediate complexity and expressive power characterise this gap, which is thoroughly analysed in this work with real-time capability and possible estimation accuracy in mind.

A lot of work has been done regarding estimations of PMSM parameters.To overcome the issue given by dependent parameters in predictive modelling and control, an incremental model is used. The conventional Model Predictive Current Control approach is used to conduct the initial study, which demonstrates that parameter inconsistency leads to incorrect predicted current values. After that, an improved incremental model is proposed[5].PMSM servo systems' speed control is improved using a predictive functional control method. The current values on the q axis are processed in the first phase using a simple model. The control law is obtained in the second phase. Because typical PFC systems perform poorly, a solution is presented that includes an Extended State Observer to mitigate the effects of disruptions.

Discusses a strategy for reducing computing time and minimising calculations in the Finite Control set-model methodology for predictive control[3]. The application is extended to PMSM drives fed by matrix converters, and the results are tested in the lab.



Process Diagram

IV. Methodology

EDA(Exploratory Data analysis)

Identify the input and target variable

A. *Target feature variables are as following*

Stator winding temperature (stator_winding)
Stator tooth temperature (stator_tooth)
Stator yoke temperature (stator_yoke)
Permanent Magnet Temperature

These are the outputs of the model which accepts the following as the input features:

Voltage q-component (u_q)
Voltage d-component (u_d)
Current d-component (i_q)
Current q-component (i_d)
Motor Speed (motor speed)
Torque (torque)
Coolant temperature (coolant)
Ambient temperature (ambient)

Data columns (total 13 columns):

#	Column	Non-Null	Count	Dtype
0	u_q	1330816	non-null	float64
1	u_d	1330816	non-null	float64
2	i_q	1330816	non-null	float64
3	i_d	1330816	non-null	float64
4	coolant	1330816	non-null	float64
5	ambient	1330816	non-null	float64
6	pm	1330816	non-null	float64
7	stator_winding	1330816	non-null	float64
8	stator_tooth	1330816	non-null	float64
9	stator_yoke	1330816	non-null	float64
10	motor_speed	1330816	non-null	float64
11	torque	1330816	non-null	float64
12	profile$_id$	1330816	non-null	int64

B. *Data visualization*

The graphical depiction of information and data is known as data visualisation. Data visualisation tools make it easy to examine and comprehend trends, outliers, and patterns in data by employing visual elements like charts, graphs, and maps.

This Dataset consists of 13 feature column, where u_q represents Voltage q component measurement in dq-coordinates (in V), coolant represents Coolant temperature (in °C), stator_winding represents Stator winding temperature (in °C) measured with thermocouples, u_d represents Voltage d-component measurement in dq-coordinates, statot_tooth represents Stator tooth temperature (in °C) measured with thermocouples, motor_speed represents Motor speed (in rpm), i_d represents Current d-component measurement in dq-coordinates, i_q represents Current q-component measurement in dq-coordinates, pm represents Permanent magnet temperature (in °C) measured with thermocouples and transmitted wirelessly via, stator_yoke represents Stator yoke temperature (in °C) measured with thermocouples, ambient represents Ambient temperature (in °C), torque represents Motor torque (in Nm) and profile_id represents Measurement session id. Each distinct measurement session can be identified through this integer id.

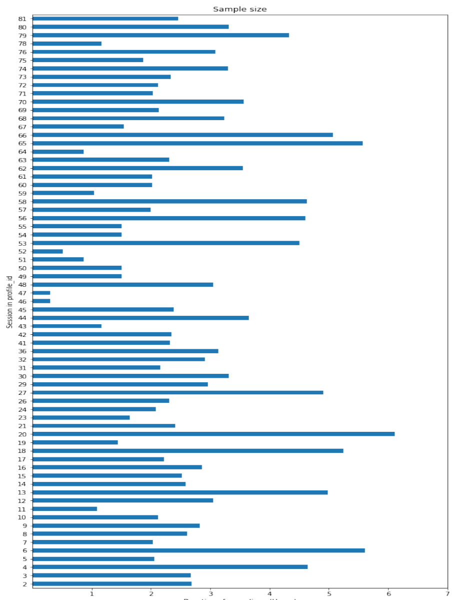


Figure 1: The data set consists of multiple measurement sessions, which can be distinguished from each other by column ”profile_id”

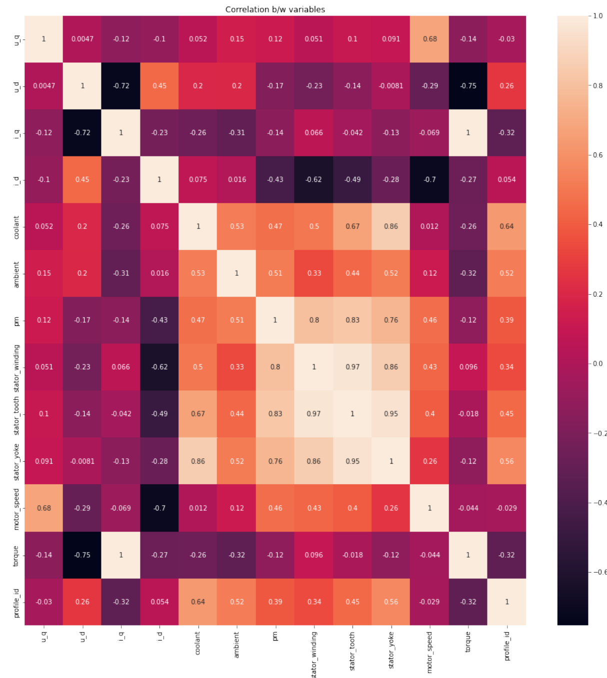


Figure 3: heat map

sample size plot shows that the data set consists of multiple measurement sessions, which can be distinguished from each other by column ”profile_id”. The plot shows that all the sessions range b/w 20 mins to around 6 hours apart from sessions ’46’ and ’47’.

As Per the ploted heatmap we can say that - stator_tooth, stator_winding and stator_yoke are correlated between them-self’s. torque has the maximum correlation coeff 1 with i_q, and strong negative correlation with u_d. motor_speed has strong positive correlation with u_q and strong negative correlation with i_d. stator_yoke is significantly correlated with coolant and less significantly with ambient. stator_tooth and stator_winding have positive correlation with coolant and negative correlation with i_d. i_q and u_d have strong negative correlation.

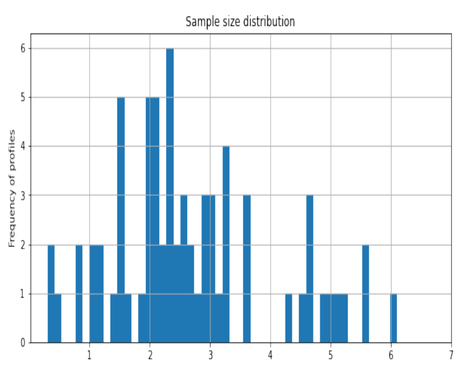


Figure 2: Sample distribution across hours of recordings

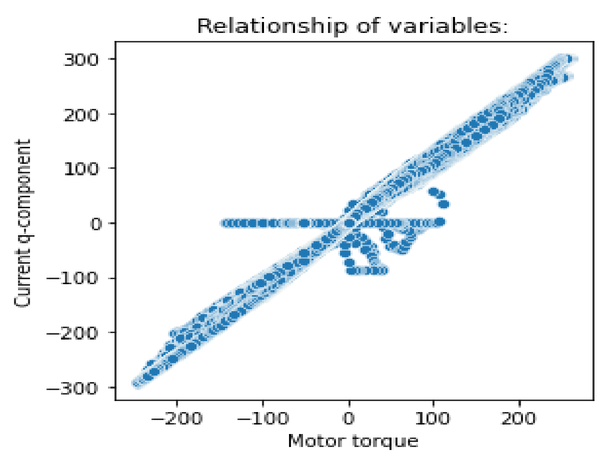


Figure 4: Relationship of variable

Daigram shows the Sample distribution across hours of recordings (number of profiles vs duration of recordings. From the figure we can see that the frequency of profile is maximum between 2-3 hours of recording.

When we plot a graph between motor torque vs current q component we can see a perfect positive correlation in which distortion is seen when motor torque and q component are at (0,0) point. Here we can see that as the value of motor torque increases we can a constant increase in q component.

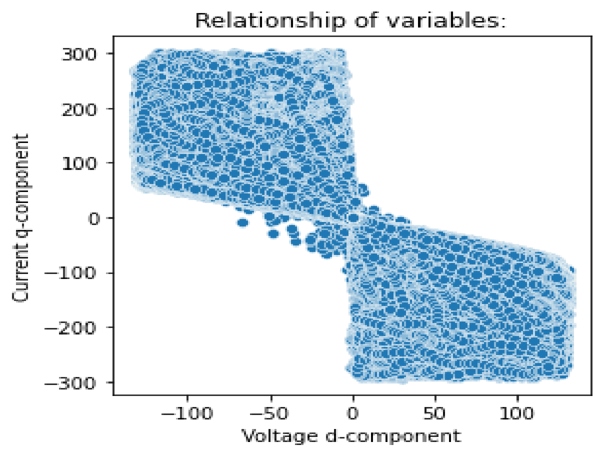


Figure 5: strong negative co relation

The figure 5 shows that i_q and u_d have strong negative correlation. When we plot a graph between voltage d-component and current q-component we can see a perfect negative correlation in which the line goes from a high value on q component down to a high value on voltage d-component.

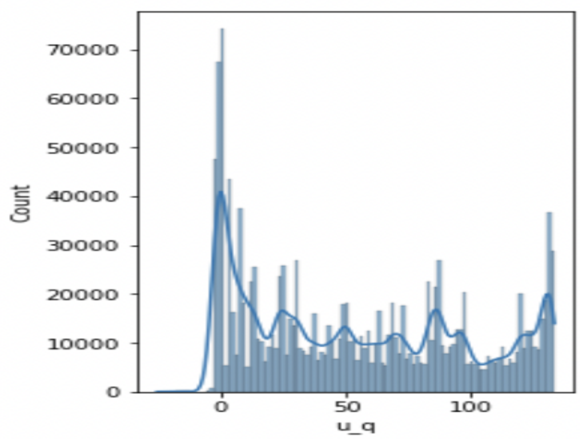


Figure 6: Distribution of independent variable

The diagram shows the Input feature 'u_q' distribution. Likewise, we have plotted the distribution graphs for all the features in the code.

C. Dataset split

We prepared two different, training sets and test sets for this project. The test set contains rows with profile ids of '65' and '72', and the data from the train set is validated in a ratio of 0.75:0.25.

1. Training set 'train df' (total 13 columns):

#	Column	Non-Null	Count	Dtype
0	u_q	1275421	non-null	float64
1	u_d	1275421	non-null	float64
2	i_q	1275421	non-null	float64
3	i_d	1275421	non-null	float64
4	coolant	1275421	non-null	float64
5	ambient	1275421	non-null	float64
6	pm	1275421	non-null	float64
7	stator_winding	1275421	non-null	float64
8	stator_tooth	1275421	non-null	float64
9	stator_yoke	1275421	non-null	float64
10	motor_speed	1275421	non-null	float64
11	torque	1275421	non-null	float64
12	profile$_i d$	1275421	non-null	int64

Above is the tabular view of the training sample. The total number of rows in this sample is 1275421. We further divided this sample into train and validation set in the ratio 0.75:0.25

1. Test set 'test df' (total 13 columns):

#	Column	Non-Null	Count	Dtype
0	u_q	55395	non-null	float64
1	u_d	55395	non-null	float64
2	i_q	55395	non-null	float64
3	i_d	55395	non-null	float64
4	coolant	55395	non-null	float64
5	ambient	55395	non-null	float64
6	pm	55395	non-null	float64
7	stator_winding	55395	non-null	float64
8	stator_tooth	55395	non-null	float64
9	stator_yoke	55395	non-null	float64
10	motor_speed	55395	non-null	float64
11	torque	55395	non-null	float64
12	profile_id	55395	non-null	int64

Above is the tabular view of the test sample. The total number of rows in this set is 55395. We splited this data sample from the original dataset with profile id's '65' and '72'.

D. Data Curation

1) Data Cleaning : The first stage in the data preprocessing section is to eliminate any unwanted columns or features that are least co-related. After analysing data visually we come to know that the column 'profile id' is not important for our model. Thus we drop the column 'profile id'. In the Data Visualization section, we have checked for missing or null values. In order to look for outliers, we also created scatter plots.

2) Extracting Target variable: We moved on to extracting Independent (input features) and Dependent (target label) variables from the training set. Here, we have 4 different target variables. So we splite the training set into two, one with input features and one with 4 traget variables. We then built 3 different machine learning models for predicting these target variables simultaneously.

3) Data Normalization: The dataset with which we are dealing with is not scaled one. We need to bring this data values into specific range in order to provide this dataset to the model. Without data normalization, the model will not produce the result which we expect. Along with that our model becomes slow and insufficient.

There are number of scaling techniques in order to scale the data. We went to several papers of this competition and found out Min-Max scalar to be the best set. The basic insight behind the Min-Max scaler is that is converts the data values in the range of 0 and 1. Thus, it is really helpful for the model to dealt with such data value. We normalize for Xtrain and Xtest data values using Min-Max scaler.

E. Model Implementation

A model implementation is the output of one or more model developers, that is technically skilled persons who translate the Abstract Risk Model and its Model Specification into computer code. Implementation may be in a variety of computer languages (or mix or languages) depending on the target IT Infrastructure. In this project we have implemented three different model Linear Regression Model, Polynomial Regressor and K-Nearest Neighbours Regressor.

1) Linear Regression Model: Regression analysis is one of the most important fields in statistics and machine learning. There are many regression methods available. Linear regression is one of them. Linear regression is one of the simplest regression method.The most significant benefit of this models is their linearity: It simplifies the estimating process and, more crucially, these linear equations have an easy-to-understand modular interpretation. Thus, we started our analysis with this basic model and compute it for these 4 target variables simultaneously. Mathematically the linear regression model equation interpreted as,

$$Y = a + bX$$

Where, Y is the dependent variable,
X is the Independent variable,
a is the intercept/constant of the line and,
b is the slope/Coefficient

2) K-Nearest Neighbour Regressor: KNN regression is a non-parametric method that, in an intuitive manner, approximates the association between independent variables and the continuous outcome by averaging the observations in the same neighbourhood.The analyst must set the size of the neighbourhood, or it can be decided using cross-validation (which we will see later) to find the size that minimises the mean-squared error. The KNN algorithm predicts the values of new data points based on 'feature similarity.' This means that a value is assigned to the new point based on how similar it is to the points in the training set. Here, KNN regression uses the same distance functions as KNN classification. The distance functions are; Euclidean, Manhattan, Minkowski. Moreover, we referred several papers related to this project and they all have seen great interest towards KNN, and the best part they even get good accuracy score also. Thus, KNN would be our most interested model for this observation.

3) Polynomial Regression: Polynomial regression is a type of linear regression that estimates the connection as an nth degree polynomial. It is a subset of Multiple linear regression. We found good accuracy using Linear Regression model for our dataset, so we jumped towards one its subset. We went thorough on the mechanism behind the Polynomial Regression model and found that it would be one of the best choice for multi-class regression. The mathematical view for the polynomial regression is shown below:

Y' = c0 + c1 x + c2 x2 ....cn xn,

Where, Y' is the predicted outcome value for the polynomial model with regression coefficients c1 to n for each degree and Y intercept c0. The model is simply a general linear regression model with n predictors raised to the power of i where i=1 to n.

V. Experimental Results and Analysis

We implemented 3 different models for this task and provided all the four target variables simultaneously to the models and perform the multi-class regression. Here, we evaluated all the models two time, first with the validation data and then with the holdout data. The results we obtained in both the evaluations stats that, result with validation data is better than with the holdout data (Obvious condition).

During our evaluation with both the data i.e, validation and holdout, we found out Accuracy score of the model, R2 score, Root mean squared error and Mean absolute error of the models. Below is the tabular view of the results we obtained during our analysis

1) Accuracy Table

algorithm	Training Accuracy	validation Accuracy	Testing accura
Lin Regr	73.73	73.800	58.956
Poly Regr	87.5462	88.0381	75.8209
KNN Regr	97.165	96.9091	49.9293

Here, when we look on the accuracy score with training and validation dataset, KNN performs much better than other two. However, for the holdout dataset Polynomial regressor has accuracy score of around 75 percent, which is much promising to our analysis.

Visualization is always the best choice for analyzing trend and pattern of each vital components of the analysis. Let's have a look at the bar chart below. The same is true for KNN's low performance on the holdout dataset. Over the taken holdout dataset, however, the polynomial regressor is the best model.

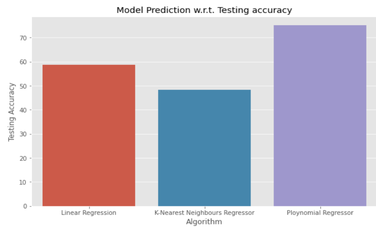


Figure 8: model prediction w.r.t Testing Accuracy

In our code, we also plotted histplots for each target variables w.r.t holdout data variables for all the three models. From this histplot we come to know, how close is the predicted value to the actual one's. Here, in the report we plotted the histplots for only our best model i.e, Polynomial Regressor. From the below histplots we can analyze that the target variables, 'Stator winding','Stator tooth' and, 'Stator york' are highly dependent on each other and has non-linear relationship with coolant temperature and ambient temperature. Because of this non-linearity we got good accuracy for polynomial regression. Whereas, for 'Permanent Magnent Temperature' we have not find any significant relationship with input variables. Due to which 'Permanent Magnet Temperature' gets predicted least better comparatively.

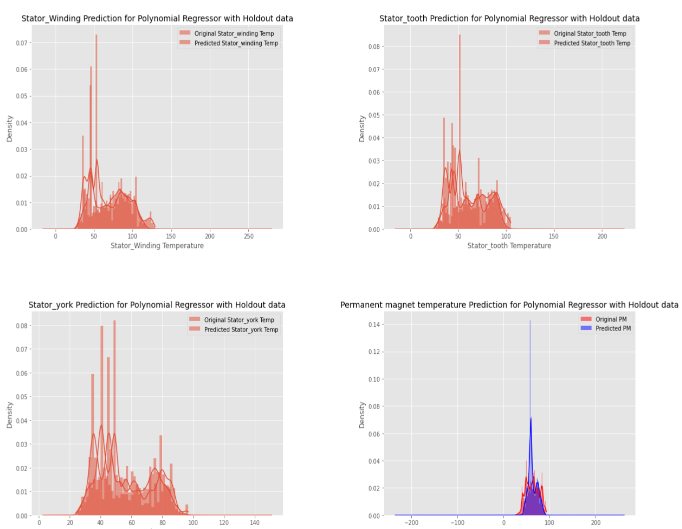


Figure 9: Visualizing the test results of Holdout dataset

Now lets Analyze our model w.r.t RMSD, MAE and R2 score:

1. Root mean square Deviation(RMSD): The standard deviation of the residuals is known as Root Mean Square Deviation (RMSD) (prediction errors). Mathametically RMSE can be depicted as,

$$\mathrm{RMSD} = \sqrt{\frac{\sum_{i=1}^{N}\left(x_i - \hat{x}_i\right)^2}{N}}$$

Figure 10: Equation of Root mean square Deviation

i = variable i
N = number of non-missing Data points
x_i = actual observations time series
x_i = estimated time series

2. Mean Absolute error(MAE): The Mean Absolute Error is a regression model evaluation metric. The mean absolute error of a model in relation to a test set is the average of the absolute values of individual prediction errors across all instances in the test set. Mathametically RMSE can be depicted as,

$$\mathrm{MAE} = \frac{\sum_{i=1}^{n}\left|y_i - x_i\right|}{n}$$

Figure 11: Equation of Mean Absolute Error

y_i = Prediction
x_i = True value
n = total number of data points

3. R2 Score: In a regression model, R-squared (R2) is a statistical measure that quantifies the proportion of variation explained by an independent variable or variables. Below is the mathematical formula to calculate R2 score,

$$R^2 = 1 - \frac{RSS}{TSS}$$

Figure 12: coefficient of determination

R^2 = coefficient of determination
RSS = Sum of Squares of Residuals
TSS = total sum of squares

In our code, for Root mean square error(RMSE) and Mean absolute error(MAE) on holdout data, Polynomial Regression performance is much better compare to Linear Regression and KNN Regression. When we talk about R2 score, again Polynomial Regression out-performs other two models. From the above results, we observed polynomial regression shows excellent outcomes. While, KNN performs really good for validation dataset, however it fails to give good accuracy with holdout data. As we know that, KNN is highly affected with even little noise and it gets overfits easily for small value of K even without noise if the dataset is large. So, to overcome this we perform hyperparameter tuning for selecting best possible values for number of neighbours and number of leafs. We also trained KNN for all the three distance metrics namely, Euclidean, Manhattan and Minkowski distance but still we did not get the significant improvement in the holdout accuracy.

The below table shows the score for RMSE, MAE and $R\hat{2}$

Model	RMS Score	MAE Score	$R\hat{2}$ Score
Linear Regr	12.65	9.90	58.62
Polynomial Regr	9.50	7.05	75.10
KNN Regr	14.14	11.55	48.31

Thus, from the overall experimental analysis, we can say that Polynomial Regression is our final model to work on and to improve it with this dataset. However, we tried so many ways to improve KNN but overfitting issue makes us unable to go through with KNN.

VI. Conclusion

A method to estimate the temperature of the permanent magnet synchronous motor has been proposed in this project. To implement this project, the algorithms of Deep learning such as Linear Regression, Polynomial Regression and KNN Regression has been used. The implementation tool is the Python programming language with add-ins such as NumPy, Pandas, Scikit-learn, Matplotlib, and Seaborn. After analysing the obtained forecasting results, we can conclude that the purpose and main objectives of this work are achieved and the trained models can be used to accurately estimate and predict the temperature of a permanent magnet synchronous motor.

VII. Future work

The scope of this work was restricted to the application of simple machine learning algorithms for a Temperature prediction of Permanent Magnet Synchronous Motor. First and foremost work that could be conducted in future would be the use of more complex Deep learning algorithms in order to predict more accurately. Secondly, due to data availability, future work could be however aim to generalize across different motor types which will require collecting data more representative of the diverse types of motors.

VIII. Acknowledgement

We want to convey our heartfelt thankfulness to God and give him all the glory for his favour, which enabled us to complete our studies successfully. We would also want to express our gratitude to our supervisor, Dr. Thangarajah Akilan, whose unwavering support and timely direction ensured that we stayed on schedule until the project was completed.

IX. References

1. W. Kirchgassner, w. oliver and b. joachim , ”Deep Residual Convolutional and Recurrent Neural Networks for Temperature Estimation in Permanent Magnet Synchronous Motors,” IEEE, p. 8, 2019.

2. A. Boglietti, A. Cavagnino, D. Staton, M. Shanel, M. Mueller, and C. Mejuto, “Evolution and modern approaches for thermal analysis of electrical machines,” IEEE Transactions on Industrial Electronics, vol. 56, no. 3, pp. 871–882, 2009.

3) F. Qi, A. Stippich, M. Guettler, M. Neubert, and R. W. D. Doncker, “Me- thodical considerations for setting up space-resolved lumped-parameter thermal models for electrical machines,” in International Conference on Electrical Machines and Systems (ICEMS), 2014.

4) O.Wallscheid, W. Kirchga ssner, and J. Bo cker. Investigation of Long Short-Term Memory Networks to Temperature Prediction for Permanent Magnet Synchronous Motors. In Proceedings of the International Joint Conference on Neural Networks, pages 1940–1947, 2017.

5) W. Kirchga ssner, O Wallscheid, and J. Bo cker. Deep Residual Convo-lutional and Recurrent Neural Networks for Temperature Estimation in Permanent Magnet Synchronous Motors. In IEEE International Electric Machines Drives Conference, pages 1439–1446, 2019.

6) W. Kirchga ssner, O. Wallscheid, and J. Bo cker. Empirical Evaluation of Exponentially Weighted Moving Averages for Simple Linear Thermal Modeling of Permanent Magnet Synchronous Machines. In Proceedings of the 28th International Symposium on Industrial Electronics, pages 318–323, 2019.