Motor Temperature Prediction with K-Nearest Neighbors and Convolutional Neural Network

Ran Le
ranle@stanford.edu

Kaijun He
hekaijun@stanford.edu

Ace Hu
acehu@stanford.edu

Abstract

Due to the complexity of the motor, direct temperature measurement of permanent magnet synchronous motor (PMSM) may not be available and the estimation through classic thermal modeling still lacks accuracy. In this report, we analyzed features according to their correlations and selected major features for models we used later to train the data. Subsequently, regression tools including simple linear regression with selected features, and with added kernels were used. We also applied classifications and convolutional neural network(CNN) before regression for motor temperature estimation considering data time dependency.In general, we are able to generate the best prediction via CNN and regression.

1 Introduction

Electric vehicles driven by permanent magnet synchronous motors(PMSM) are becoming more and more popular. Many experimental studies have shown a drop in vehicle performance as well as thermal robustness at high motor temperatures, but sensor-based temperature measurement such as infrared thermography[1] is technically and economically infeasible due to the sophisticated motor structure and the difficult accessibility of the rotor. As a result, indirect measurements are taken and the internal temperature is calculated using classic thermal models, which requires expert domain knowledge and could hardly be transferred to general cases.

Our task is to use machine learning and deep learning tools learnt in class to make better prediction on motor temperature.The result can be applied to real-life situation in optimizing motor performance and giving out warnings when the motor temperature is abnormally high.

The input to our algorithm are sensor data collected from PMSM deployed on a test bench and the output are temperatures of different components. In order to output temperature estimations, we performed experiments using linear regression model as the baseline and added kernel features to reduce bias. Considering collected data are time dependent, we also performed K nearest neighbor regression by first classifying data with nearest data points and applying regression afterwards. CNN was experimented to train the data by first dividing data into time series and was found to yield the best estimation over all.

2 Related Work

Research on motor temperature predictions were previously based on basic heat transfer theories. These models purely rely on motor parameters without using any testing data. Later on, a gray-box model was introduced called lumped-parameter thermal networks (LPTNs) which needs both physical thermal parameters and real test bench data[2]. These methods require knowledge of related fields to model the system and carefully choose suitable parameters for modelling.

As a substitution, black-box approaches using testing data were investigated and a slightly weaker level of estimation accuracy was achieved, which proved the concept of data-based modeling[3]. Afterwards, estimations using Deep Residual Convolutional and Recurrent Neural Networks was investigated and proved to be accurate[4]. Moreover, to maintain less model parameters and reach similar accuracy, linear regression models were investigated. However,

Preprint. Under review.

temperature sequences exhibit strong correlation along neighboring observations, which has not been exploited in previous research yet[5].

Regression following training data in CNN by Modukuru [6] was considered a good method for prediction by us, because it is able to make accurate estimations and CNN performs well in analysis of sensor data in time series. However, the result is not always stable and not deterministic. It took time to find parameters such as learning rate to yield the best result. In our investigation, we also trained data through CNN, but with kernel features added. One dropout layer after convolutional layer was also added to prevent co-adaptation of feature detector. MSE in this case is able to converge in shorter period of time and in a more stable way.

3 Dataset and Features

Our model was trained on the Electric Motor Temperature dataset which contains 1 million data collected from different testing sessions on the same PMSM with 2Hz sampling rate[7]. The data has inputs including ambient temperature, voltage, motor speed, etc. and outputs including temperature measurements at different components. The data were taken from different measurement sessions which are unique and independent from each other.

By looking at the data, output are pm, $stator\_yoke$, $stator\_tooth$, $stator\_winding$, which represent PMSM rotor temperature, stator yoke temperature, stator tooth temperature, and stator winding temperature.

In order to reduce dimensions for the training, we first find out the most influential components by investigating the correlation between the input features and the outputs. Thus, the correlations were figured out and shown by a heat map in Fig. 1. Features with average absolute correlation to outputs over 0.2 were first considered as the main features for training: $ambient$, $coolant$, i_d, $motor\_speed$, which represent ambient, coolant temperatures, current d component and motor speed.

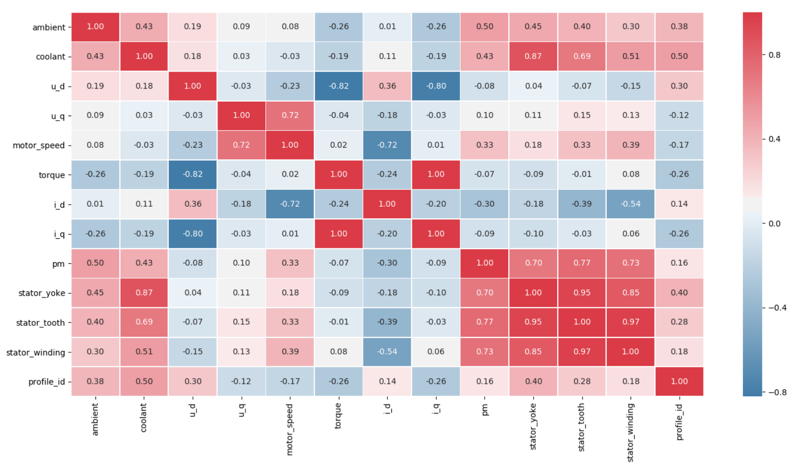

Figure 1: Heat map of all parameters.

In linear models and k-nn model, we separate data randomly in 3:7 to test set and train set. In CNN, data was separated into time sequences with m samples in a sequence, input is m-dimensions and output is the target temperature after the time sequence ends.

4 Methods

Based on the results from previous researches and related works, the team selected three models for further investigation. Linear regression with kernels is used as a baseline, K-nearest neighbors regression will be tested and compared with other regression methods, and finally, time sequence will be added to the convolutional neural network.

4.1 Linear Regression

To predict specific values for output instead of classifying, linear model was first implemented for the simplicity. To briefly explain the algorithm, we assumed there are n data and each with P features and the prediction y. Denote the a single input vector $\mathbf{X} = [\mathbf{x}_1, \mathbf{x}_2, \cdots \mathbf{x}_P]$. The estimated output in linear model [5] is

$$\hat{y}^{(i)} = \theta_0 + \sum_{p=1}^{P} x_p^{(i)} \theta_p. \tag{1}$$

Then the residual sum of squares (RSS)[5] is the loss function to be minimized by finding the optimal $\theta = [\theta_0, \theta_1, \cdots, \theta_P]$:

$$RSS(\theta) = \sum_{i=1}^{n} (y^{(i)} - \hat{y}^{(i)})^2. \tag{2}$$

Input parameters $\mathbf{X_i}$ were selected by calculating correlation [8] between each input and output. Input parameters with high correlation were selected as features in our model.

$$corr(X, Y) = \frac{E[XY] - E[X]E[Y]}{\sqrt{E(X^2) - E(X)^2}\sqrt{E(Y^2) - E(Y)^2}} \tag{3}$$

Some features have solid physical meanings when related such as the magnitudes of current and voltage are defined as the vector summation of DC and AC parts, and the power is defined as the product of current and voltage.[5] Thus, they could be the kernel terms and these new features were engineered to extend the feature space for machine learning.

$$i = \sqrt{i_d^2 + i_q^2}, \quad u = \sqrt{u_d^2 + u_q^2}, \quad P = ui \tag{4}$$

4.2 K-nearest neighbors regression

K-nearest neighbors is an algorithm making predictions based on similarity measurements. Considering the time dependency of the data and the fact that all input data are continuous, we used K-nearest neighbor algorithm [9] to first classify the data points by looking for nearest neighbors c_i with Euler distance for m features:

$$d = \sqrt{\sum_{i=1}^{m} (x_i - c_i)^2} \tag{5}$$

and then predict values with the sum of weighted neighbors regarding to the inverse of Euler distance:

$$output = \sum_{i=1}^{k} \frac{1}{d} y_i \tag{6}$$

4.3 Convolutional neural network

Data is divided into time sequences, and the goal is to predict temperature after time series. The reason of using CNN is that it can learn from the time series data. Furthermore,the network structure consists a 1-D convolutional layer with kernel size same as size of input in a time series, a dropout layer, and two fully connected layers with 100 neurons. Adding the dropout layer can make the learning process slower but more stable. Activation function used is conventional Relu function. Network structure 2 is shown below:

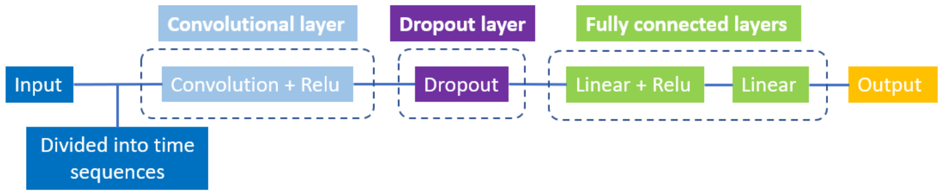

Figure 2: CNN and Regression structure

5 Experiments, Results and Discussion

We used three methods mentioned above to train the models and evaluated the model with the test set.

5.1 Linear Regression

Our baseline is to use main features found above to perform linear regression. Mean Squared Error(MSE) for all outputs are shown in Table 1.

Then we considered adding features with physical meaning as indicated in Eq. 4. We got smaller error as shown in Table 1. The density plot for actual and predicted $stator\_winding$ with and without kernel features are shown in Fig.

Linear Regression Model	pm	$stator\_yoke$	$stator\_tooth$	$stator\_winding$
Without Kernel features	0.5286	0.2729	0.2230	0.1812
With Kernel features	0.3487	0.1846	0.1495	0.1229

Table 1: MSE for Linear Regression models

3 and 4. From both the table and figures, we found adding kernels feature indeed improved accuracy. Therefore, we decided to include kernel features to other models.

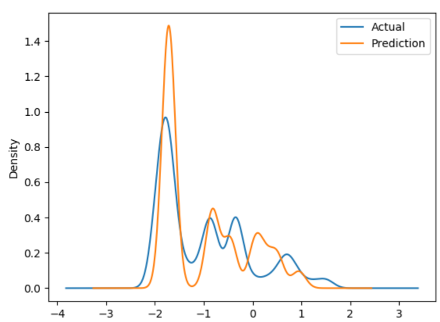

Figure 3: Density plot of actual and simple linear model prediction output $stator\_winding$.

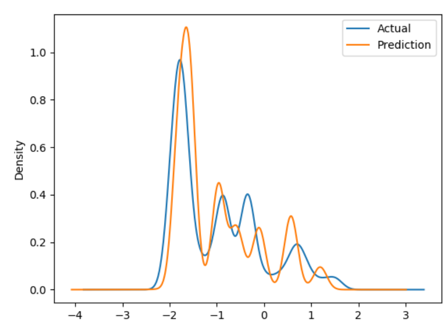

Figure 4: Density plot of actual and linear with kernel feature prediction output $stator\_winding$.

Features used for models below are: $ambient, coolant, motorspeed, i, u$, which represent ambient, coolant temperatures, motor speed, RMS current, and RMS voltage.

5.2 K-nearest neighbors regression

For K-nearest neighbors regression, we implemented the model for k=1 (one nearest neighbor) up to k=20 (20 nearest neighbors), and found that the best result with smallest Root Mean Squared Error(RMSE) yielded at k=7. Table 2 below shows the RMSE errors regarding to different outputs and k-values from 5 to 8:

k-values	pm	$stator\_yoke$	$stator\_tooth$	$stator\_winding$
k=5	0.1967	0.0927	0.1259	0.1363
k=6	0.1970	0.0923	0.1256	0.1360
k=7	0.1967	0.0920	0.1254	0.1362
k=8	0.1972	0.0920	0.1255	0.1364

Table 2: RMSE for K-nn with different k-values

We use RMSE here to better distinguish differences between different k values. Compared with results from linear regression model, MSE for pm when k=7 is 0.0387, which is much smaller and therefore K-nn performs better in prediction of motor temperature.

The reason for the improvement could be due to the classification performed in the K-nn model before regression. As we know that data collected are heavily dependent on time, only considering small-range of local points while making prediction is reasonable.

5.3 Convolutional Neural Network and regression

For the best performance on CNN, we tested the model on different learning rate and number of data points in a time sequence. Finally we found MSE=0.000720 on test set is smallest when learning rate=0.02, using 7 samples in a time sequence, 5 in batch size. These parameters are experimented and able to give us a fast, stable convergence of MSE. Following figures are MSE of training set on CNN (Fig. 5) and prediction on randomly selected test samples (Fig. 6). It is evitable that MSE is able to converge in several iterations, and the threshold of difference here is set to be 1×10^{-6}.

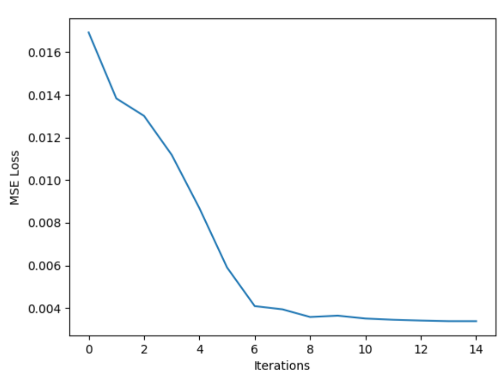

Figure 5: Training loss in CNN

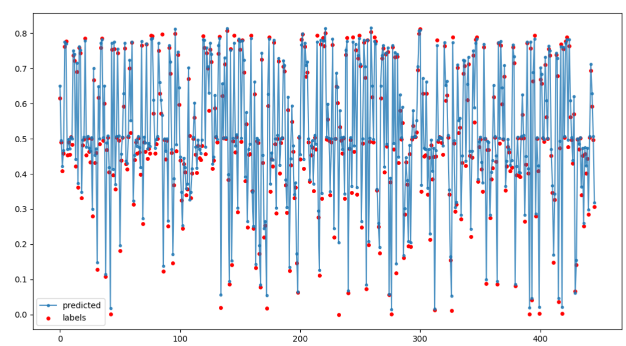

Figure 6: Prediction and real labels pm on test set

6 Conclusions and Future work

It is shown that with careful feature selection and feature engineering, it is possible to precisely predict motor temperatures with different machine learning tools. K-nearest neighbor method performs much better than linear regression because of the locally weighted estimation on nearest neighbors, by considering the time dependency Among all the tested models, regression with CNN performs the best after data divided into time sequences. The time sequence is investigated and proved to be an important component for making predictions. However, we have to mention due to the fact that CNN is stochastic, different experiments may lead to different results.

Although the models are proved to be efficient on testing PMSM, we have not checked the feasibility of transferring the model to other types of motors. Future works could focus on model adaptation and validate the accuracy.

7 Contributions

The team worked closely together for planning and implementing algorithms throughout the project and each team member was responsible for specific parts. Ran focused on data analyzing, feature engineering and kernel selection. Kaijun mainly worked on machine learning algorithms and testing. Ace was responsible for convolutional neural network algorithms, and model results analysis. Special thanks for the help from CS 229 teaching assistant Leo, Andrey and Jonathan on useful advises.

8 GitHub Link

https://github.com/hhhkkkjjj/CS229_proj.git

References

[1] Martin Ganchev, Bernhard Kubicek, and Hansjörg Kapeller. Rotor temperature monitoring system. pages 1 – 5, 10 2010.

[2] O. Wallscheid and J. Böcker. Global identification of a low-order lumped-parameter thermal network for permanent magnet synchronous motors. *IEEE Transactions on Energy Conversion*, 31(1):354–365, March 2016.

[3] O. Wallscheid, W. Kirchgässner, and J. Böcker. Investigation of long short-term memory networks to temperature prediction for permanent magnet synchronous motors. In *2017 International Joint Conference on Neural Networks (IJCNN)*, pages 1940–1947, May 2017.

[4] W. Kirchgässner, O. Wallscheid, and J. Böcker. Deep residual convolutional and recurrent neural networks for temperature estimation in permanent magnet synchronous motors. In *2019 IEEE International Electric Machines Drives Conference (IEMDC)*, pages 1439–1446, May 2019.

[5] W. Kirchgässner, O. Wallscheid, and J. Böcker. Empirical evaluation of exponentially weighted moving averages for simple linear thermal modeling of permanent magnet synchronous machines. In *2019 IEEE 28th International Symposium on Industrial Electronics (ISIE)*, pages 318–323, June 2019.

[6] Naga Sai Pranay Modukuru. Cnn time series regression - motor temperature, Sep 2019.

[7] Electric motor temperature, Jun 2019.

[8] Alexander Craig Aitken. Statistical mathematics. 1949.

[9] Naomi S Altman. An introduction to kernel and nearest-neighbor nonparametric regression. *The American Statistician*, 46(3):175–185, 1992.