Capstone Project

Niranjan Salimath

Machine Learning Engineer Nanodegree

July 23nd, 2019

Definition

Project Overview

In most automotive applications, a high power and torque density as well as efficiency is required for any given motor, which makes the permanent magnet synchronous machine (PMSM) the preferred choice. In order to exploit the machine's full capabilities, high thermal stress on the machine's potentially failing components must be taken into account. A sensor-based temperature measurement would yield rather precise knowledge regarding the machine's thermal state, yet for the rotor part, it is technically and economically infeasible due to an electric motor's sophisticated internal structure and the difficult accessibility of the rotor. Hence, direct rotor monitoring techniques such as infrared thermography [1], [2] or classic thermocouples with shaft-mounted slip-rings [3] fall short of entering industrial series production. In contrast, stator winding temperature monitoring is measured on a sensor basis nowadays, yet in case of faults, these sensors cannot be replaced due to being firmly embedded in the stator. In addition, sensor functionality may deteriorate during the motor's life cycle. Taking into account the ever increasing importance of functional safety especially in the automotive industry, redundant temperature information becomes obligatory.

In this project, it will be shown that a supervised machine learning models, namely linear regression and support vector regression circumvents any kind of electric motor modeling by being fitted on measured test bench data with minor preprocessing directly and still achieves high estimation performance with time invariant properties. This approach ignores domain knowledge and, hence, does not require any conventional engineering expertise in electric machine thermal management.

Problem Statement

The goal is to create a algorithm to predict the rotor temperature in permanent magnet synchronous machines.

Permanent magnet synchronous machines (PMSMs) are a popular choice in many traction drive applications due to their high energy and power density and moderate assembly costs. However, electric motor thermal robustness in general is harmed by the lack of accurate temperature monitoring capabilities such that safe operation is ensured through oversized materials at the cost of its effective utilization. Classic thermal modeling is conducted through lumped parameter thermal networks (LPTNs), which help to estimate internal component temperatures rather precisely but also require expertise in choosing model parameters and lack physical interpretability as soon as their degrees of freedom are curtailed in order to meet the real-time requirement. It will be shown that, as an alternative to LPTNS, supervised learning algorithms like linear regression, logistic regression and support vector regressor achieve similar predictive performance with low computational complexity as long as input representations are preprocessed with exponentially weighted moving averages. Thus, domain knowledge becomes neglectable, and estimation performance depends entirely on collected data and considered input representations.

The tasks involved are the following:

1. Download and preprocess the Kaggle's electric-motor-temperature dataset (pmsm_temperature_data.csv)
2. As input data for this project is a time series, whole of the data needs to be visualized, trained and tested by grouping.
3. Identify important features which are influencing the rotor temperature by visualizing the input data using "matplotlib" or "seaborn".
4. Train regressor algorithms to predict rotor temperature.
5. Calculate evaluation metrics. And Compare with benchmark model.
6. Use advanced methods like ensembling methods to improve accuracy.

Evaluation Metrics

The evaluation metric to be used is mean squared error to quantify the performance of both the benchmark model and the solution model.
If y_i is the predicted value of the i-th sample, and y_i is the corresponding true value, then the mean squared error (MSE) estimated over $n_{samples}$ is defined as

$$\text{MSE}(y, \hat{y}) = \frac{1}{n_{\text{samples}}} \sum_{i=0}^{n_{\text{samples}}-1} (y_i - \hat{y}_i)^2.$$

Analysis

Data Exploration

Kaggle Datasets link: pmsm_temperature_data.csv.

Dataset of Electric-Motor-Temperature Project (https://www.kaggle.com/wkirgsn/electric-motor-temperature)

Input fields of data:

1. Ambient – Ambient temperature as measured by a thermal sensor located closely to the stator.
2. Coolant - Coolant temperature. Motor is water cooled. Measurement is taken at outflow.
3. u_d - Voltage d-component
4. u_q - Voltage q-component
5. motor_speed - Motor speed

6. torque - Torque induced by current.
7. i_d - Current d-component
8. i_q - Current q-component
9. pm - Permanent Magnet surface temperature representing the rotor temperature. This was measured with an infrared thermography unit.
10. stator_yoke - Stator yoke temperature measured with a thermal sensor.
11. stator_tooth - Stator tooth temperature measured with a thermal sensor.
12. stator_winding - Stator winding temperature measured with a thermal sensor.
13. profile_id - Each measurement session has a unique ID. Make sure not to try to estimate from one session onto the other as they are strongly independent.

Data organization:

ambient	coolant	u_d	u_q	motor_speed	torque	i_d	i_q	pm	stator_yoke	stator_tooth	stator_winding	profile_id
-0.75214	-1.11845	0.327935	-1.29786	-1.2224282	-0.250182	1.029572	-0.24586	-2.52207	-1.8314217	-2.0661428	-2.0180326	4
-0.77126	-1.11702	0.329665	-1.29769	-1.2224293	-0.249133	1.029509	-0.24583	-2.52242	-1.8309687	-2.0648587	-2.0176313	4
-0.78289	-1.11668	0.332772	-1.30182	-1.2224278	-0.249431	1.029448	-0.24582	-2.52267	-1.8304	-2.064073	-2.0173435	4
-0.78094	-1.11676	0.3337	-1.30185	-1.2224301	-0.248636	1.032845	-0.24695	-2.52164	-1.8303328	-2.0631368	-2.0176322	4
-0.77404	-1.11678	0.335206	-1.30312	-1.2224286	-0.248701	1.031807	-0.24661	-2.5219	-1.8304977	-2.0627947	-2.0181448	4
-0.76294	-1.11695	0.334901	-1.30302	-1.2224286	-0.248197	1.031031	-0.24634	-2.5222	-1.8319309	-2.0625494	-2.017884	4

Target feature: Rotor temperature [“pm”]

Exploratory Visualization

As the input data is a time series data, it needs to be grouped before splitting it into training set and testing set. Inside every group data needs to be split into training set and testing set. In this problem, as “profile_id” column indicates the different durations during which data is captured, hence we can group the data based on “profie_id”.

Before jumping to splitting data, if we see at individual group lengths[number of time stamps taken in particular profile_id] we can rule out some of the groups as they don’t carry much of the information regarding temperature change.

Measurement session lengths

Fig 1 shows that all measurement sessions range from 20 minutes to around 6 hours. The two short session ids "46" and "47" might be not very representative as temperatures inside electric motors need time to vary.

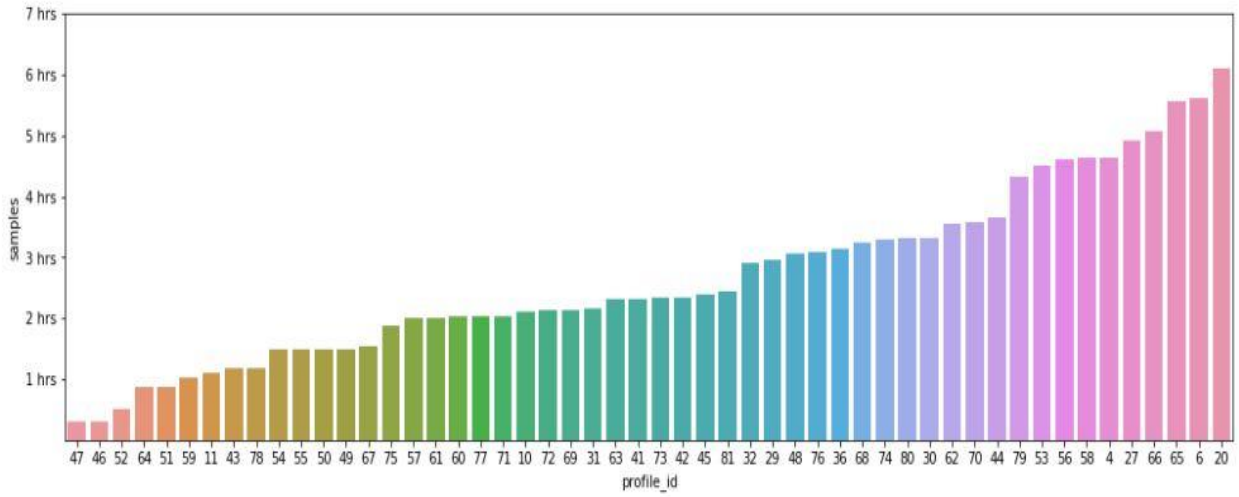


Fig 1

Effect of some uniformly picked recordings to select target features.

In real motor rotor temperature will mostly follow stator temperatures. Hence our target features will be stator_yoke, stator_winding, stator_tooth and pm(rotor temperature). Here temperatures of stator at different part of it are considered as target features as they will be resembling the rotor temperature with less deviation.

Other features like torque, motor_speed and coolant temperature will also be influencing the rotor temperature.

Visualization of the above mentioned features are captured in the Fig 2

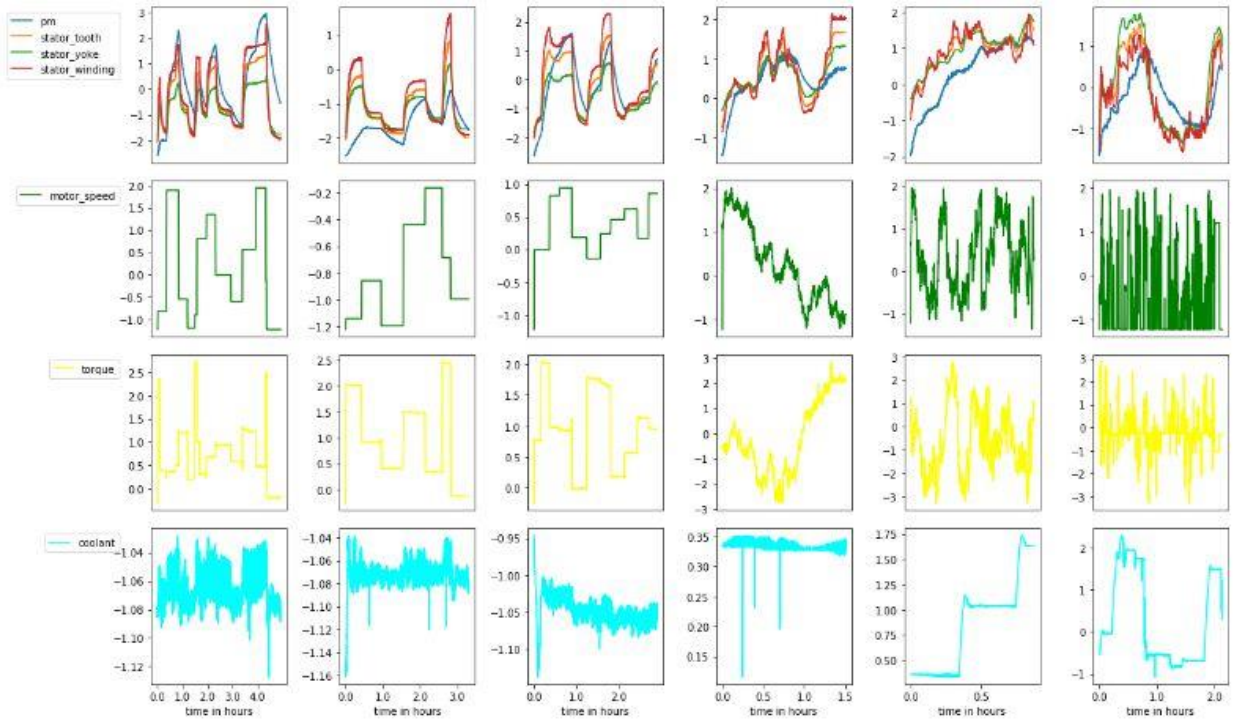


Fig 2

Findings from the above visualization

1. While motor excitations (motor_speed, torque, coolant) are sometimes of high dynamic, sometimes of stepwise nature, target temperatures always exhibit low-pass behavior with exponential rise and falls.
2. Coolant temperature suffers from measurement artefacts expressed by sharp drops in temperature, which recover as fast.
3. PM (Permanent Magnet -> Rotor) temperature expresses the slowest time constant and follows stator temperatures

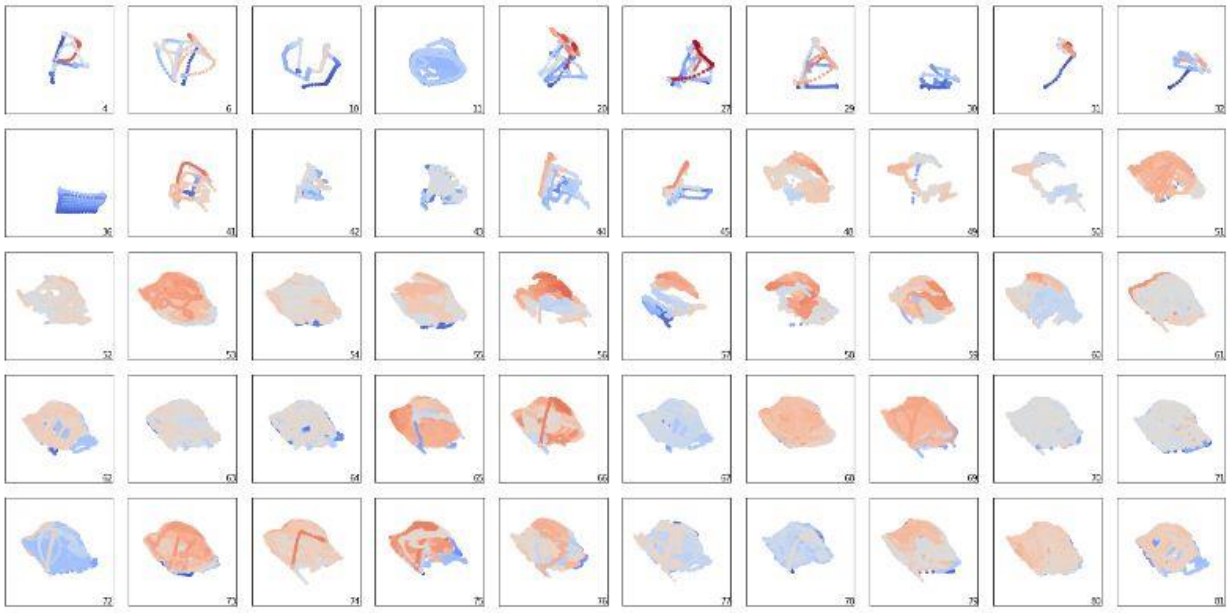


Fig 3

Finding from Fig 3:

It is obvious that lower profile IDs are of simpler driving cycles, not moving much in feature space. Higher profile IDs are driving cycles of high dynamics - excitation happened through random walks in the motor_speed-torque-plane.

Algorithms and Techniques

Algorithm used is Support vector regressor[SVR] with 'linear' kernel. As SVR takes more time to fit floating point values, I used "MinMaxScaler ()" from sklearn's preprocessing library. MinMaxScaler will convert all floating values to values between 0 and 1 for default case.

Make pipeline function is used for applying MinMaxScaler to data feeded to SVR and rename that particular configuration with different name.

For improvement of accuracy and reducing Mean squared error I used Random forest regressor as ensembling method with 10 estimators.

Benchmark

To create an initial benchmark I used Linear regression model which is a very simple model but catches linearity in data. It gave me the mean squared error value of 0.3558. which is very poor in this case with keeping random test ids like 58 and 78.

Methodology

Data Preprocessing

Grouping of input data based on their profile Ids:

As this dataset is a time series and the data in different time series cannot be used for predicting the rotor temperature in different time series, I grouped all data points based on their profile IDs.

Splitting of data into training set and testing set is also done in all groups making sure that every group has both training and testing set.

Principal component analysis [PCA]:

PCA is used for extracting principal features which are affecting change in rotor temperature. Which is used for implementation of benchmark model linear regression model.

MinMaxScaler is used to scale input data to support vector regression model to reduce time consumption for fitting the model. As randomly distributed floating values takes a lot of time while fitting to SVR model.

Implementation

Implementation of the project starts from benchmark model linear regression model. For linear regression model I used principle component analysis as a data preprocessor. Linear regression model yielded me mean squared error of 0.3558 which is very high in this case.

In new implementation with support vector regression model with linear kernel is very good in handling both variance and overfitting / underfitting. But one major problem with this model is it consumes more time if the input is random floating values.

Hence to avoid this problem I used MinMaxScaler from sklearn's preprocessing library. Which scales down all the floating values between 0 and 1 in default case. Hence the time consumption reduces drastically.

Steps involved in implementation:

1. Load the input data into pandas data frame and remove NAs.
2. Identify important features which are influencing the rotor temperature by visualizing the input data as mentioned in data visualization section.
3. Transform the input data using Principal component analysis technique for benchmark model.
4. Split whole data into both the training and validation set in every group of data, preprocessing them as described in the data preprocessing section.
5. Initialize the SVR model with "linear" kernel.
6. Fit the model.

Refinement

As mentioned in benchmark model the accuracy was very poor with mean squared error metric, I used SVR model which is good in performance when we compare with linear regression model.

But Even then mean squared error for SVR model is also not good enough in real time. Hence in the next step I used Ensemble method to reduce mean squared error and increase it's relevance in real time.

Here I used Random forest as a ensemble method which uses Decision trees as estimators. And is real good at controlling overfitting of decision trees. I have set n_estimators to 10 for my calculation.

Results

Model Evaluation and Validation

Final model built is Randomforest regressor.

Randomforest regressor model has least Mean squared error when we compare with benchmark model [Linear regression] and Support vector regressor.

Time consumed in fitting benchmark model is very less but mean squared error is very high. In the same way for support vector regressor also time consumed is less than a minute [Run on 16GB RAM Windows 10 PC with 2.4GHz frequency] and mean squared error is also less compared to linear regression model.

In random forest regressor model time consumed in fitting is approximately 2 minutes but mean squared error has been reduced drastically.

For best model for a particular dataset, mean squared error should be very low with acceptable amount of time consumption.

Justification

The Benchmark model(linear regression model) has a mean squared error of 0.3558 , while the final model has mean squared error of 7.232393517095166e-09. [i.e. 0.000000007232393517 09….] which is very good for this dataset.

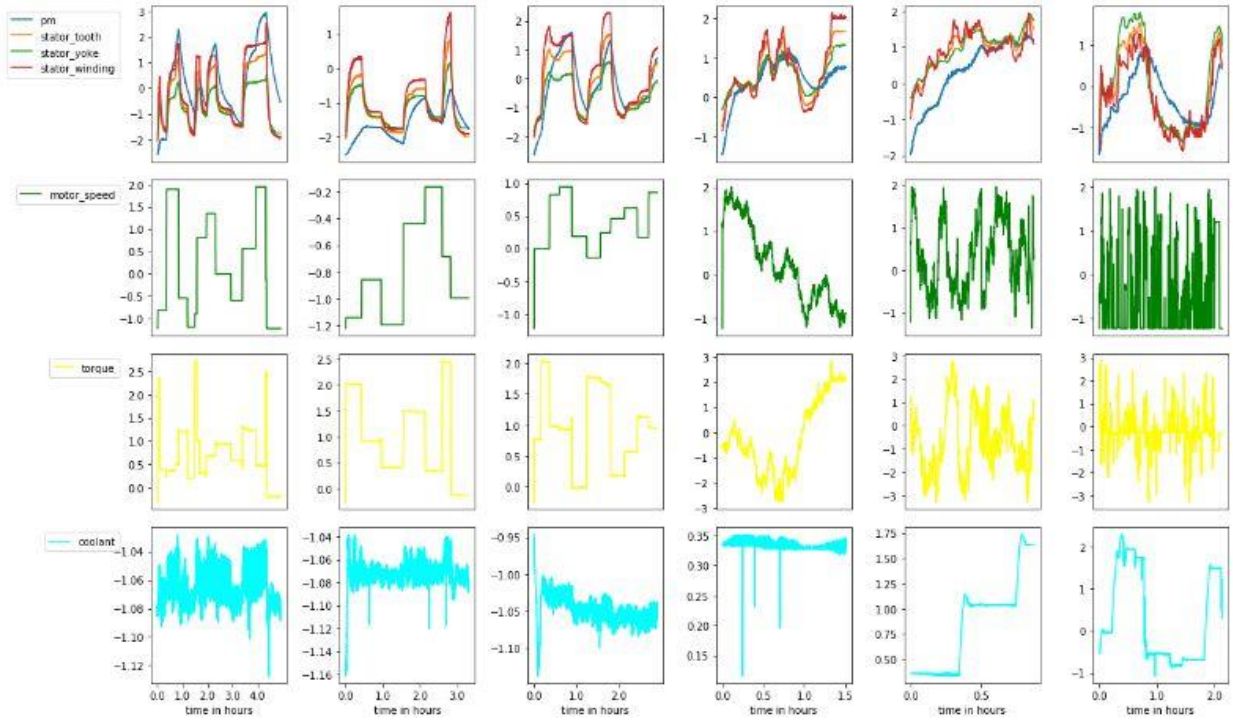
So, clearly final model is stronger model.

I think the model is significant enough for this problem’s solution.

Random forest regression model is a robust bagging ensemble method with very low mean squared error for time series dataset.

Conclusion

Free-Form Visualization



Important quality of the project is the target feature “pm” (rotor temperature) which is highly dependent on other features like temperatures of stator_winding , stator_yoke and stator_tooth. If we see first row of graphs it is clear that rotor temperature is gradually following temperatures of above mentioned features.

Rotor temperature is exhibiting very slow time constant.

Reflection

The process used for this project can be summarized using the following steps:

1. Get data from Kaggle. The dataset is in csv format so, convert it to a pandas dataframe.
2. As input data for this project is a time series, whole of the data needs to be visualized, trained and tested by grouping the dataset based on profile IDs.
3. Identify important features which are influencing the rotor temperature by visualizing the input data using “matplotlib” or “seaborn”.
4. Selected one benchmark model as linear regression model.
5. Train benchmark model to predict rotor temperature.
6. Calculated evaluation metrics. And saw it’s criticality on problem statement.
7. Tried with powerful model support vector regression model and repeated steps 5 and 6.
8. For better performance I used ensemble method Random forest regression which robust with very good performance on time series dataset.
9. Repeated steps 5 and 6.
10. Finally I came to conclusion that random forest regression model is efficient enough for this problem.

Improvement

I used number of estimators for final model i.e. random forest regression as 10. With increasing number of estimators and tuning some parameters of the final model we can reduce mean squared error to very low value.