Capstone 2 Final Report:
Electric Motor Temperature Prediction

1.0 Problem Statement

The world is transitioning from cars that run on fossil fuels to cars that run on electricity. Governments have demonstrated a strong commitment to incorporating the electrification of cars as a key component to meet net zero targets. Recent announcements include:

- An executive order in the United States in August 2021, which set a new ambition for electric vehicles (EVs) to represent 50% of light duty vehicle (LDV) sales in 2030.
- In October 2021, an announcement of ambitions to have 100% zero-emission LDV sales by 2035 in Chile.
- In Canada a new target is to achieve 100% zero-emission LDV sales by 2035 instead of 2040. New interim targets of 20% zero-emission LDV sales by 2026 and 60% by 2030 were also established.

Increasingly, automakers have been exploring electrification plans to both comply with policy regulations and maintain a competitive position in a transitioning market (IEA, 2022).

This leads to the discussion of electric motor performance. Rotor temperature and torque are not reliably and economically measurable in a commercial vehicle. Being able to have strong estimators for the rotor temperature helps the automotive industry to manufacture motors with less material and enables control strategies to utilize the motor to its maximum capability. Furthermore, a precise torque estimate leads to more accurate and adequate control of the motor, reducing power losses and eventually heat build-up (Kirchgässner et al., 2021).

The goal of this project is to precisely predict the torque and temperature of an electric motor given other sensor measurements.

2.0 The Dataset

To achieve this goal, a dataset from Kaggle will be used. The dataset comprises several sensor data collected from a permanent magnet synchronous motor (PMSM) deployed on a test bench. The PMSM represents a German OEM's prototype model. Test bench measurements were collected by the LEA department at Paderborn University.

All recordings are sampled at 2 Hz (once every 0.5 seconds). The data set consists of multiple measurement sessions, which can be distinguished from each other by column *profile_id*. A measurement session can be between one and six hours long.

The motor is excited by hand-designed driving cycles denoting a reference *motor_speed* and a reference *torque*. Currents in d/q-coordinates (columns *i_d* and *i_q*) and voltages in d/q-coordinates (columns *u_d* and *u_q*) are a result of a standard control strategy trying to follow the reference speed and torque. Columns *motor_speed* and *torque* are the resulting quantities achieved by that strategy, derived from set currents and voltages.

Most driving cycles denote random walks in the speed-torque-plane in order to imitate real world driving cycles to a more accurate degree than constant excitations and ramp-ups and -downs would.

The dependent variables of this investigation that we want to be able to predict are: *stator_winding*, *stator_tooth*, *stator_yoke*, *pm*, and *torque*.

The dataset is a single csv file containing all measurement sessions and features, totaling 185 hours of motor runtime. Table 1 highlights the features of the dataset and their definitions, below:

Table 1: Sensor data definitions

Sensor Variable	Definition
profile_id	Measurement session id. Each distinct measurement session can be identified through this integer id
u_q	Voltage q-component measurement in dq-coordinates (in V)
u_d	Voltage d-component measurement in dq-coordinates
i_d	Current d-component measurement in dq-coordinates
i_q	Current q-component measurement in dq-coordinates
motor_speed	Motor speed (in rpm)
coolant	Coolant temperature (in °C)
ambient	Ambient temperature (in °C)
stator_winding	Stator winding temperature (in °C) measured with thermocouples
stator_tooth	Stator tooth temperature (in °C) measured with thermocouples
stator_yoke	Stator yoke temperature (in °C) measured with thermocouples
pm	Permanent magnet temperature (in °C) measured with thermocouples and transmitted wirelessly via a thermography unit
torque	Permanent magnet temperature (in °C) measured with thermocouples and transmitted wirelessly via a thermography unit torque Motor torque (in Nm)

3.0 Data Wrangling

The dataset can be loaded from its csv format to a Pandas Dataframe, where its values can then be easily inspected. The data contains no *None* values, has no missing values, and has appropriate value types for each column. The dataset does not require any wrangling past the creation of the Pandas Dataframe to begin exploratory data analysis.

4.0 Exploratory Data Analysis

Given that the dataset contains the voltages and currents being used to excite the motor and their resulting sensor values, the correlation of the features is a good place to start. Figure 1, below, shows a heat map (with a diverging color palette) of the Pearson correlation coefficient between each feature.

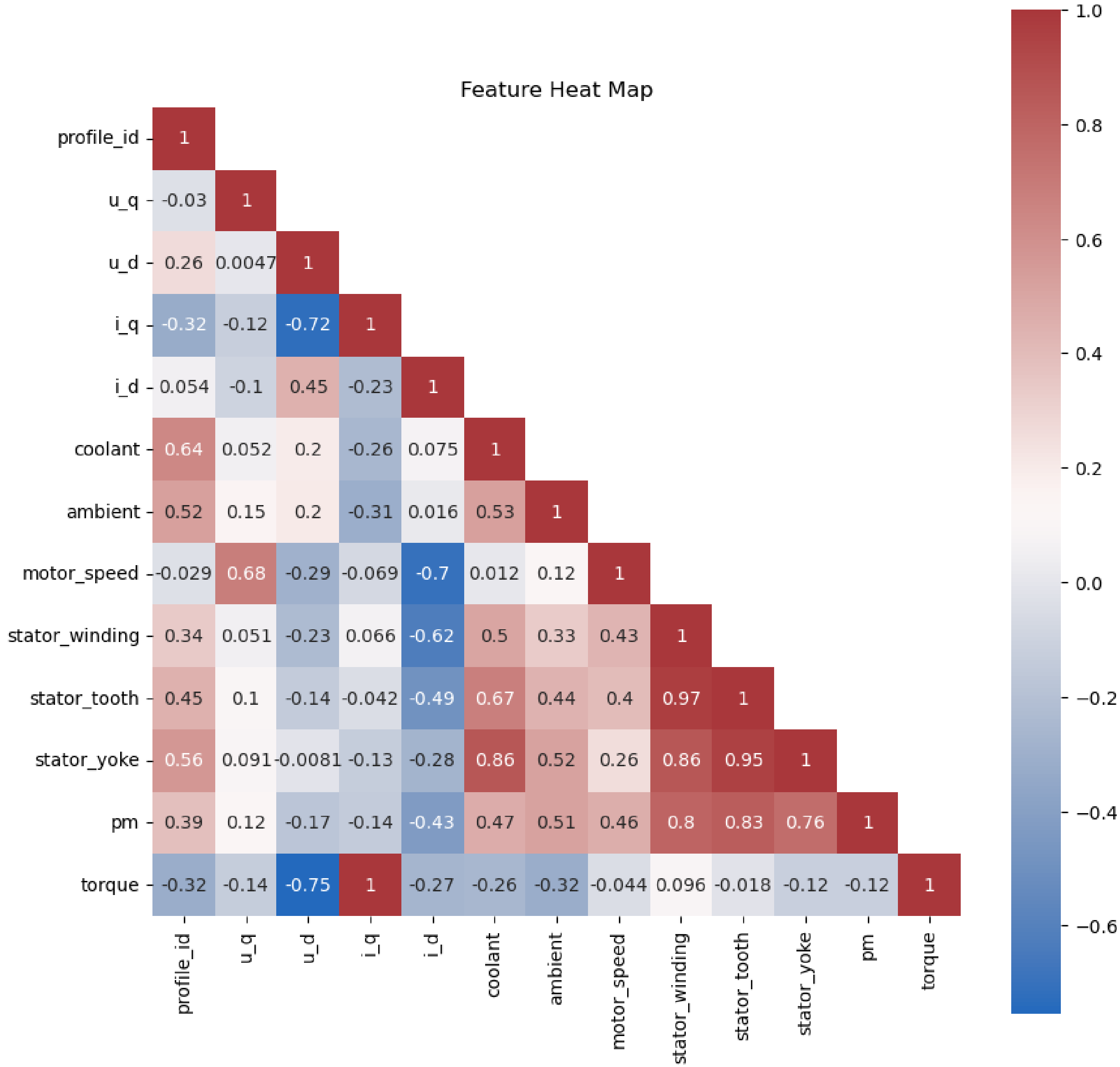

Figure 1: Heat map representing the Pearson correlation coefficients between features

From a first impression:

- Currents (*i_d* and *i_q*) and voltages (*u_d* and *u_q*) appear to be correlated, as expected, given Ohm's law (Britannica, T. Editors of Encyclopaedia, 2023).
- *motor_speed* and *torque* seem to correlate with currents (*i_d* and *i_q*) and voltages (*u_d* and *u_q*), as expected, since current and voltages are the inputs of the motor control strategy.
- The *stator_yoke*, *stator_tooth*, *stator_winding*, and *pm* temperatures seem to be correlated with one another.

Now to visualize these discovered correlations, we can see two examples below, in Figure 2. To prevent overcrowding the plots, only three measurement sessions (*profile_id*) are shown at a time:

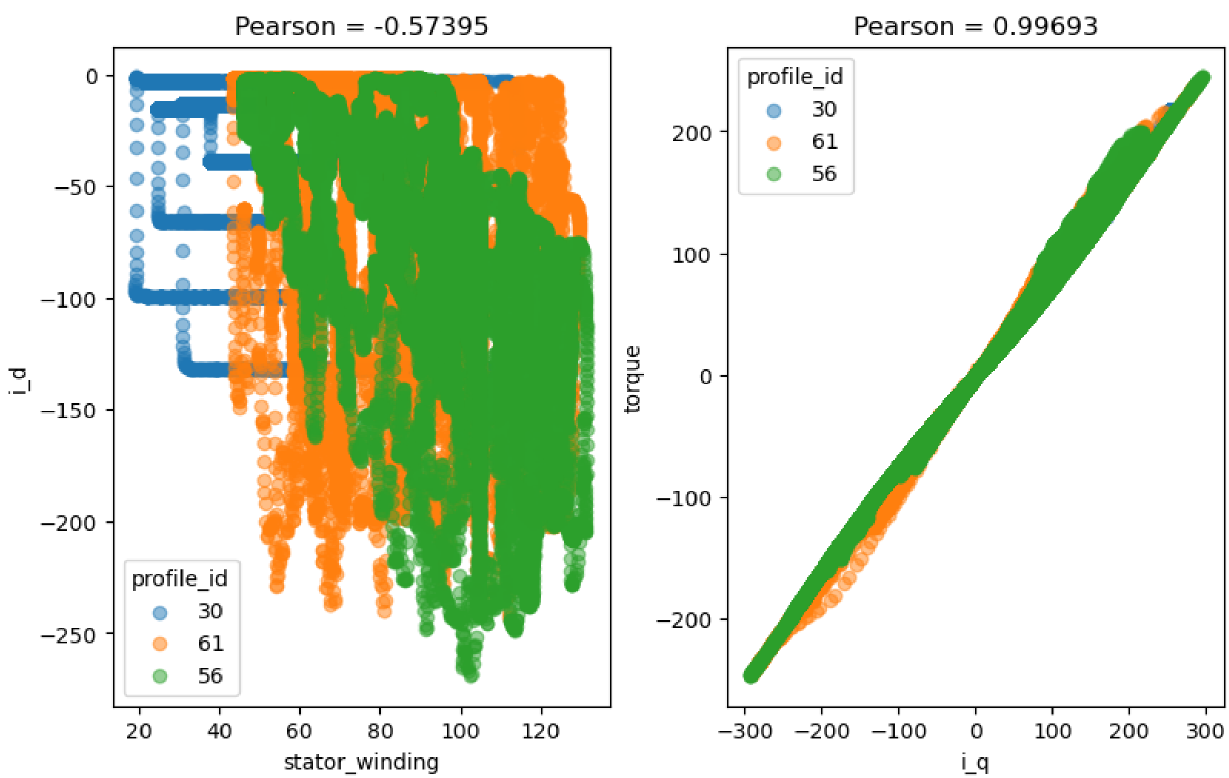

Figure 2: Visualization of feature correlations

The high correlation (0.997) between *i_q* and *torque* is clearly pictured in the right plot. This feature to feature correlation will be useful when it comes to model development. However, the ambiguous correlation between *i_d* and *stator_winding*, or more generally, between current and motor temperature is not proving to be useful.

Since the sensor measurements in this dataset are recorded over time, it is important to understand what this time-series data looks like over the course of the different measurement sessions. Again, only three sessions are being plotted at a time, below, in Figure 3:

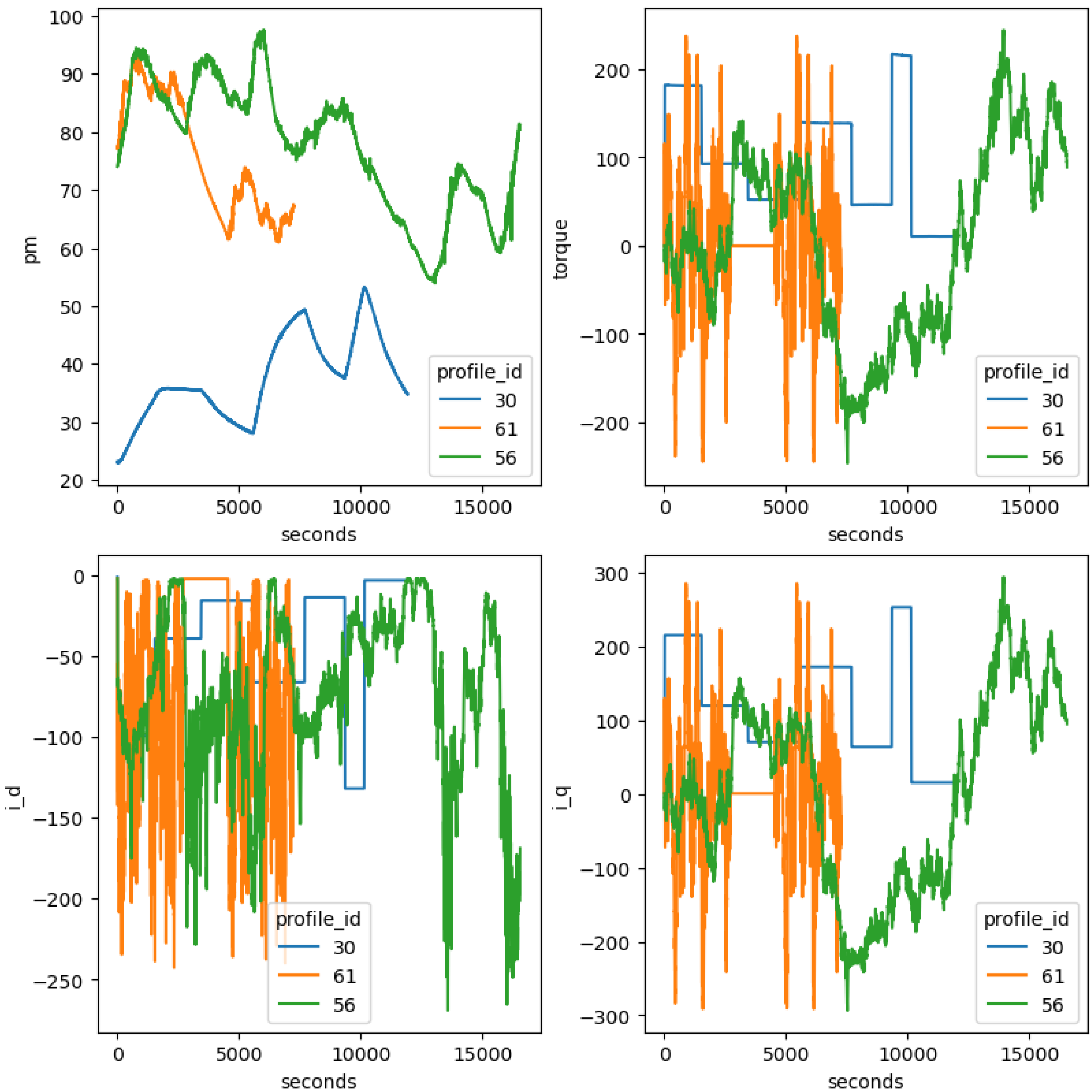

Figure 3: Time-series visualization

From the time-series visualization above, it can be seen that the control inputs for each *profile_id*, vary in duration as well as pattern. The extent to how well variables *torque* and *i_q* correlate can be seen here. Motor temperature, seen in *pm*, appears to drift in value over time as well as have a more complicated relationship with the motor control strategy.

When it comes to exploring models to predict our target features, we know that simple linear regression should perform well for the torque prediction.

5.0 Preprocessing and Training Data Development

For preprocessing, a *runtime* variable was created to begin at zero seconds for each *profile_id* and increase in increments of 0.5 seconds, to match the recording of the data at 2 Hz.

To eventually compare predictive models, a training dataset and a testing dataset will need to be created from the original dataset. To avoid test data leakage into the training dataset, the train-test split should be created before scaling. Two different methods for creating the train-test split are:

1. The *profile_id* sessions, as a whole, can be split into train and test sets.
2. Segments of each *profile_id* time-series can be split into train and test sets.

As a particular *profile_id* demonstrates a new motor cycle, it makes more sense to treat a subset of the *profile_id* sessions as the training dataset and the remaining *profile_id* sessions as the testing dataset. This is because after a predictive model is deployed into production it will have to treat all new motor cycles as "test" datasets.

That being said, the different measurement sessions will be randomly sampled to create a training set of *profile_id* sessions and a testing set of *profile_id* sessions with a 75-25 split.

Now that testing data is separate from the training data, the "features" and "targets" can be defined. From this point on, the "features" will denote the variables in the dataset being used to predict the "targets". The "targets" are the variables in the dataset that we want to predict. Both the training and testing datasets are split into these "feature" and "target" datasets denoted by X_train, X_test, y_train, and y_test, where X is for features as y is for targets.

The next step is to fit scikit learn's StandardScaler to the training dataset features. After fitting the scaler, it can be used to transform both the training and testing datasets' features. Note that the targets are not being scaled.

6.0 Modeling

Five different regression models were tested when determining a final model. These models include:

1. Linear regression
2. Ridge regression
3. Lasso regression
4. Polynomial regression
5. Exponentially Weighted Moving Average regression

Linear regression was the obvious starting point after viewing the highly correlating feature and target variables *i_q* and *torque*, respectively. Ridge and Lasso regression were explored to see how the linear model could benefit by shrinking the model coefficients towards, or to, zero. Polynomial regression was explored to see if any thought-to-be linear relationships were better modeled by a higher degree. Lastly, Exponentially Weighted Moving Average (EWMA) regression was explored due to the recommendation of a research paper that highlights that permanent magnet synchronous motor (PMSM) temperatures can be modeled efficiently by the use of exponentially weighted moving averages (Kirchgässner, Wallscheid, and Böcker).

Two metrics are used to compare models: the coefficient of determination (R^2) and the root-mean-squared-error ($RMSE$).

6.1 Linear Regression

Scikit Learn's Linear Regression model was used. Note that this model has no hyperparameters to tune. To illustrate how each measurement session prediction scored from the test dataset for a given model, boxplots of the measurement session metric scores are shown for each target variable. Figure 4, below, shows how this model's predictions scored for each target variable, according to the R^2 statistic:

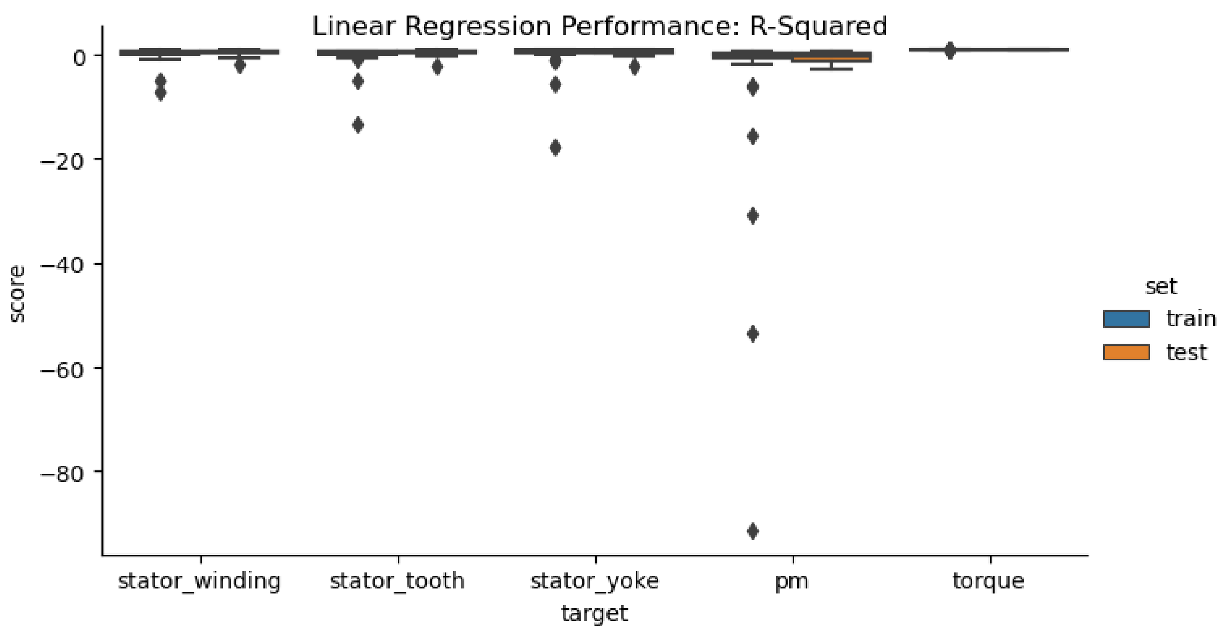

Figure 4: Linear regression R^2 scoring

Note that the four temperature target variables have instances where a linear regression model scored negative R^2 values. This is possible when the predictive model is predicting so poorly that the prediction error is worse than the null error: the mean. So it is clear that a linear model does not predict motor temperature well at all. Figure 5, below, shows the $RMSE$ for each target variable in the same manner:

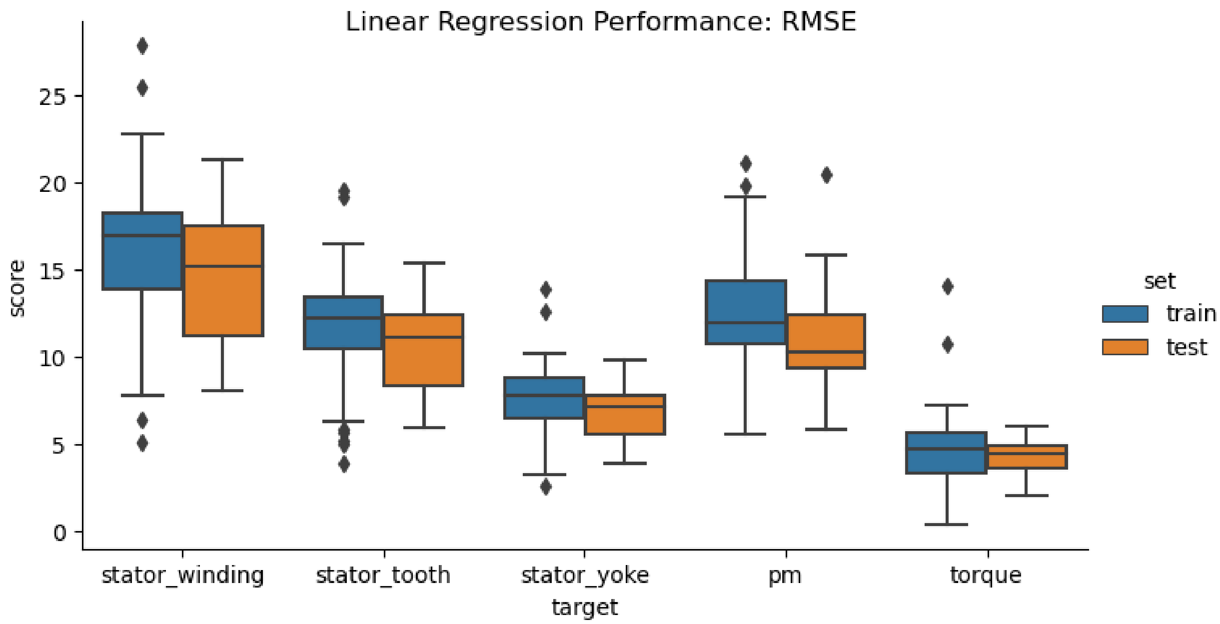

Figure 5: Linear regression $RMSE$ scoring

The *torque* variable has the lowest error out of the target variables, which agrees with the results of the R^2 metric for this model. This metric will be used to compare with the other models rather than look at individually.

6.2 Ridge Regression (L2 Regularization)

Ridge regression is L2 regularization and shrinks the magnitude of fitted model coefficients towards zero, reducing multicollinearity. This model has one hyperparameter: alpha (α) to tune. Alpha was tested in a gridsearch from 1e-10 to 1e10 with 5-fold cross validation. The results are shown below in Figure 6:

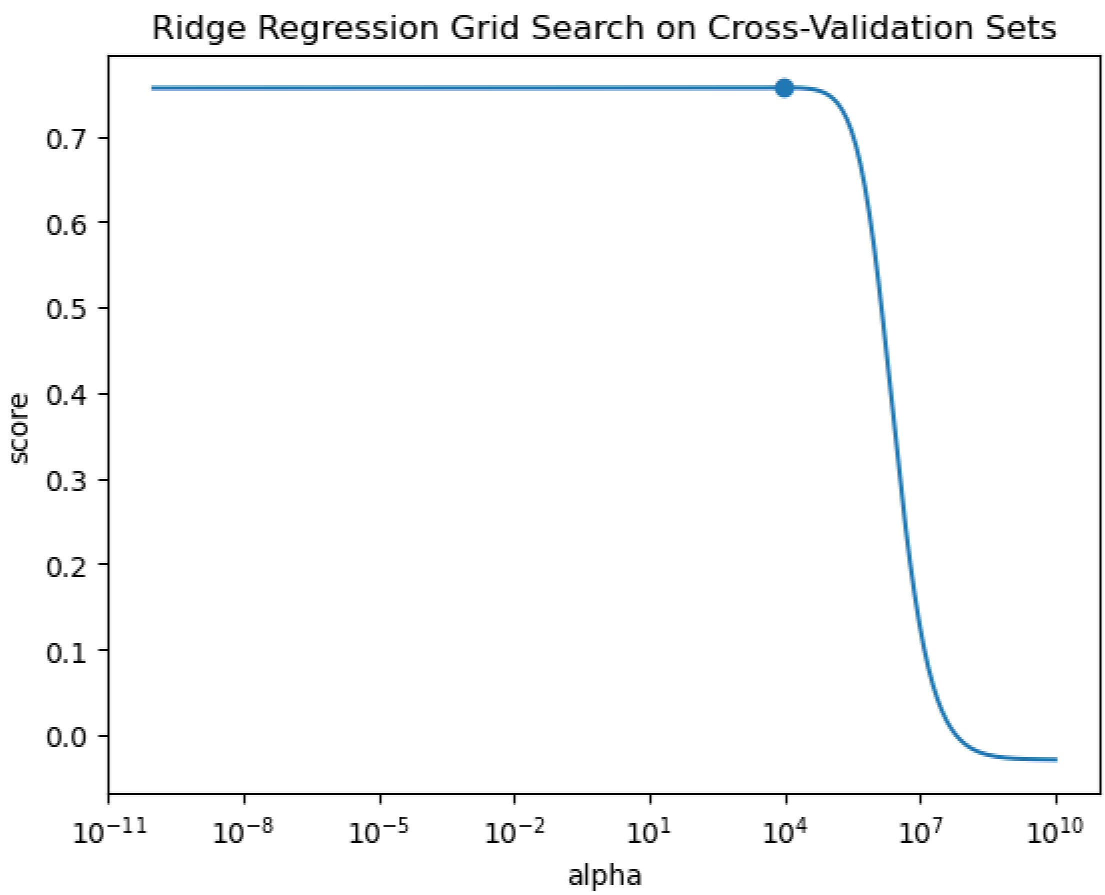

Figure 6: Ridge regression grid search results on cross-validation sets

The best scoring alpha value (alpha = 9329.3) is denoted by the dot on the above plot. When it came to the model scoring, using this optimal value of the alpha parameter, there wasn't much change in the R^2 scoring. Figure 7, below, highlights the results:

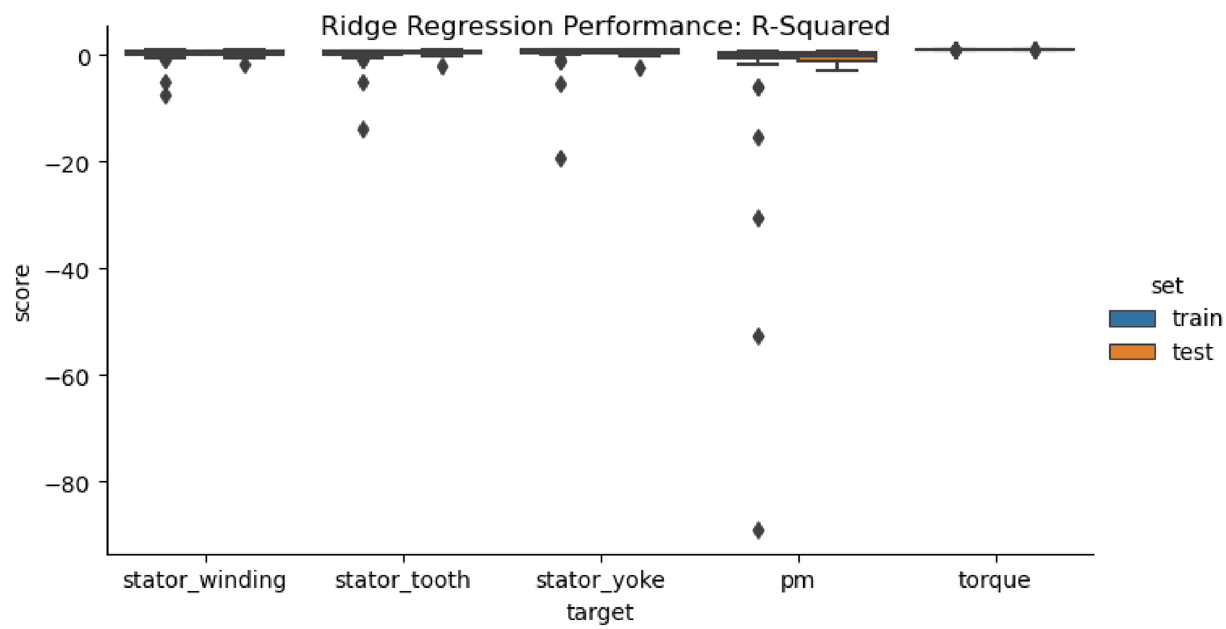

Figure 7: Ridge regression R^2 scoring

The temperature target variables are still receiving a negative score, meaning that ridge regression is not going to be the model to use for temperature prediction. For comparison, the $RMSE$ scoring is viewable, below, in Figure 8:

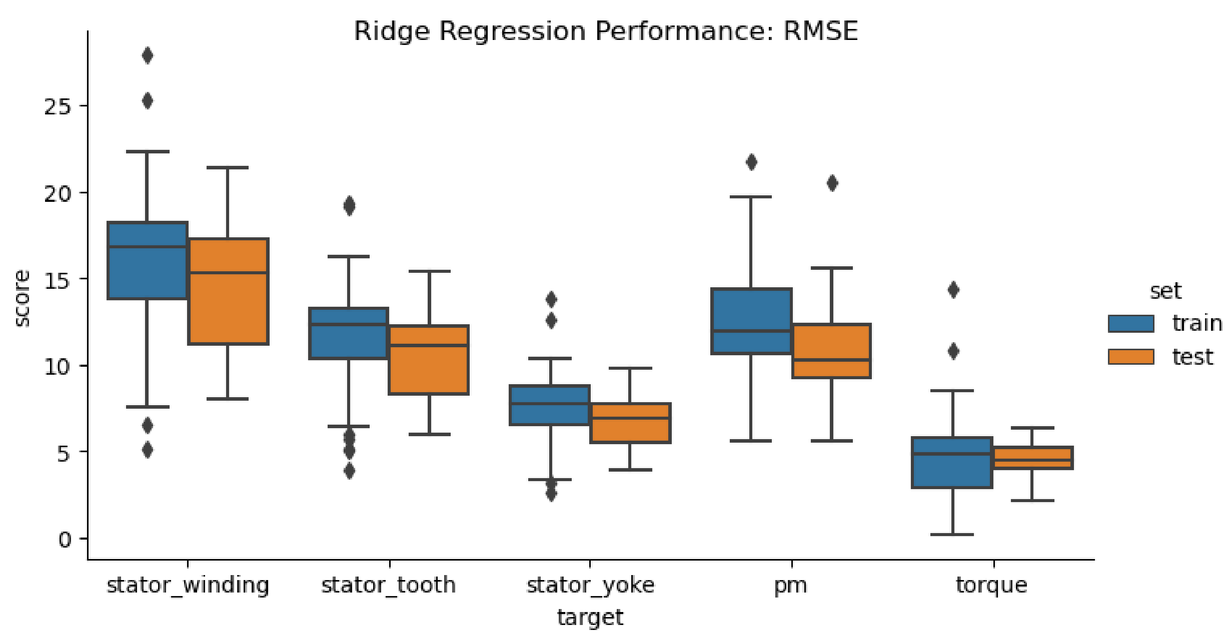

Figure 8: Ridge regression $RMSE$ scoring

To view how the Ridge regression is shrinking the coefficients towards zero, the following plot shows the coefficients as a function of alpha, as well as the best alpha denoted as the dotted line, in Figure 9:

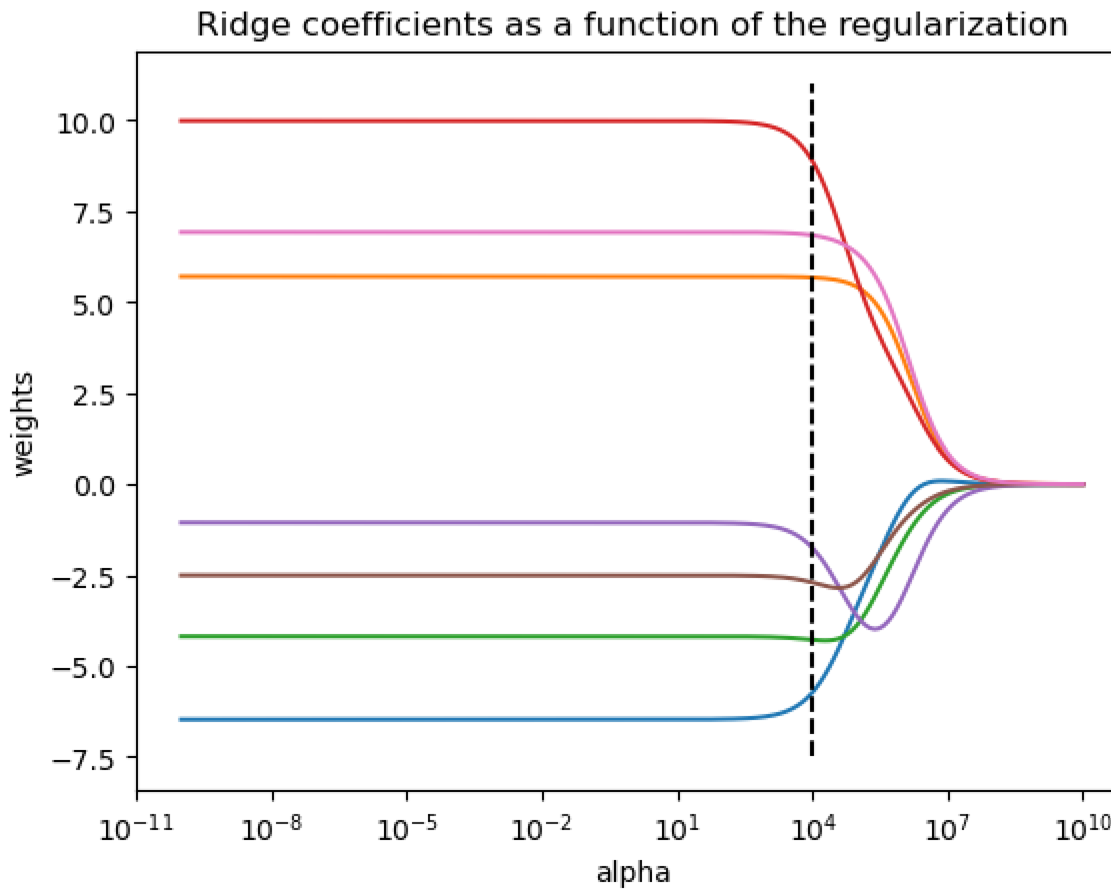

Figure 9: Ridge coefficients as a function of the regularization

It can be seen that the optimal value of alpha does result in some shrinkage of the coefficient weights. However, the Ridge regression does not produce prediction results that we are aiming for.

6.4 LASSO Regression (L1 Regularization)

LASSO (Least Absolute Shrinkage Selector Operator) regression automatically does feature selection by shrinking coefficients to zero. LASSO regression also has only one hyperparameter to tune: alpha. A gridsearch from 1e-10 to zero with 5-fold cross validation was conducted and the results are shown below in Figure 10:

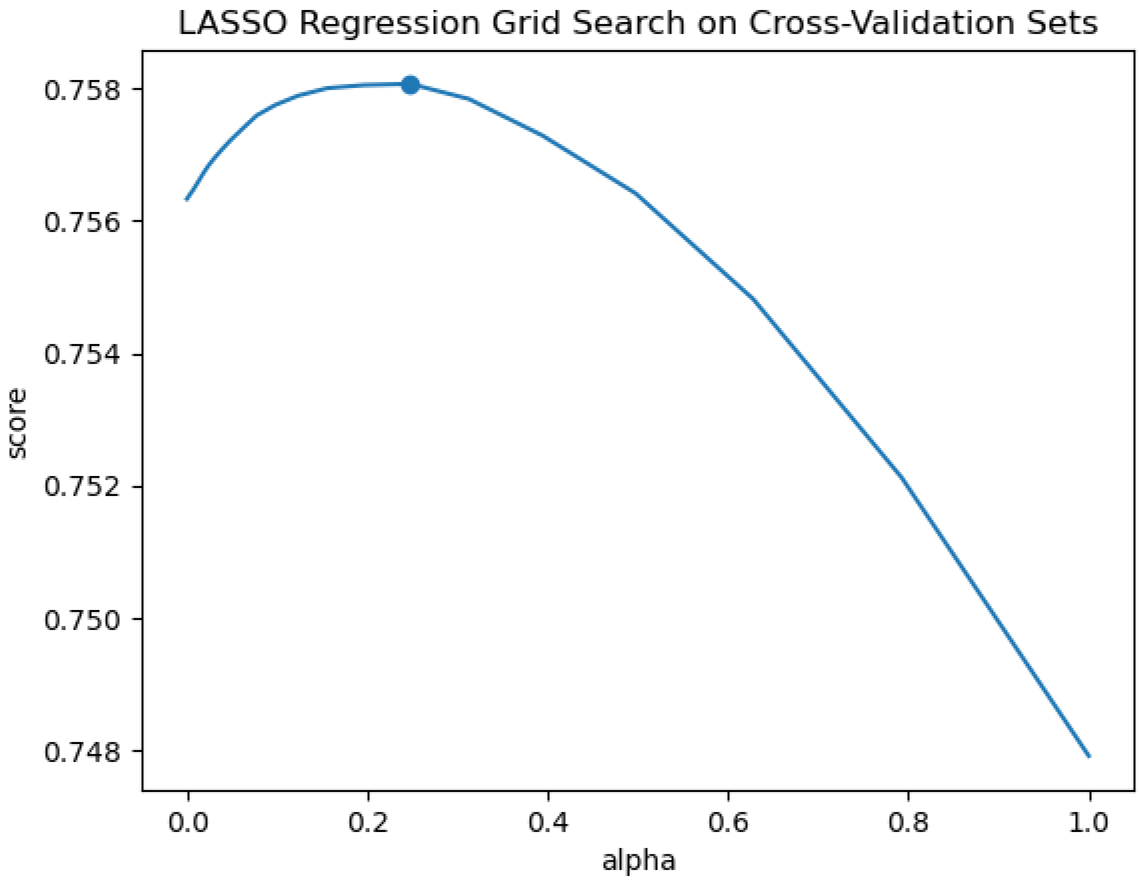

Figure 10: LASSO regression grid search results on cross-validation sets

The best scoring alpha value (alpha = 0.248) is denoted by the dot on the above plot. When it came to the model scoring, using this optimal value of the alpha parameter, negative R^2 scoring is still prevalent. Figure 11, below, highlights the results:

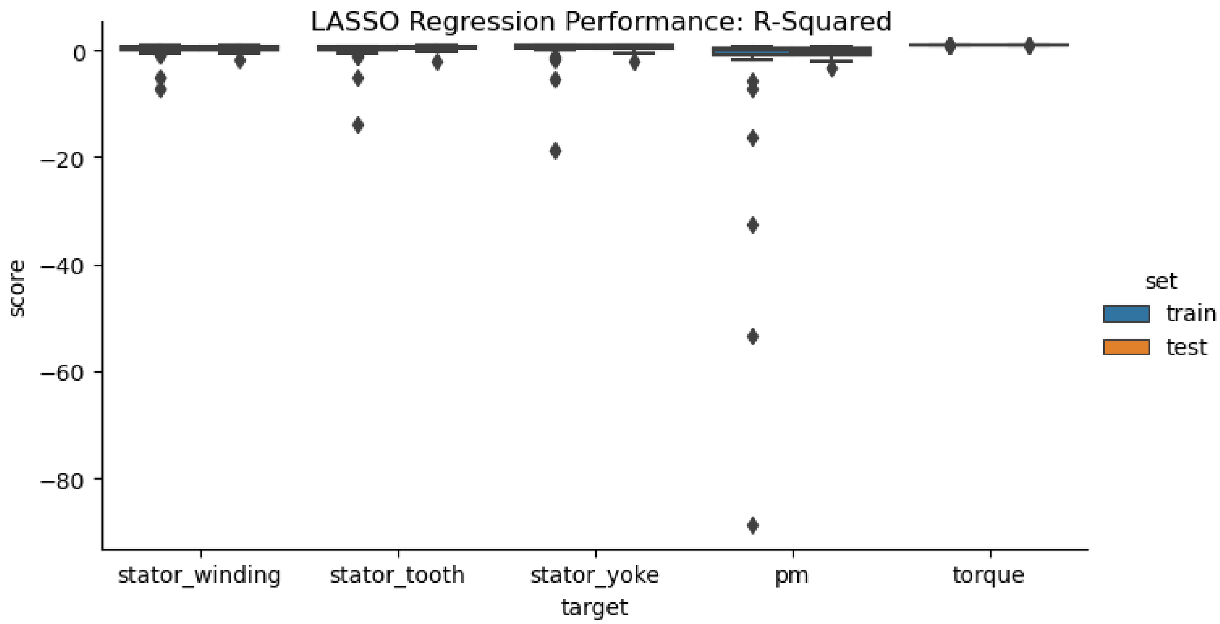

Figure 11: LASSO regression R^2 scoring

The error results for the LASSO regression model predictions are below, in Figure 12:

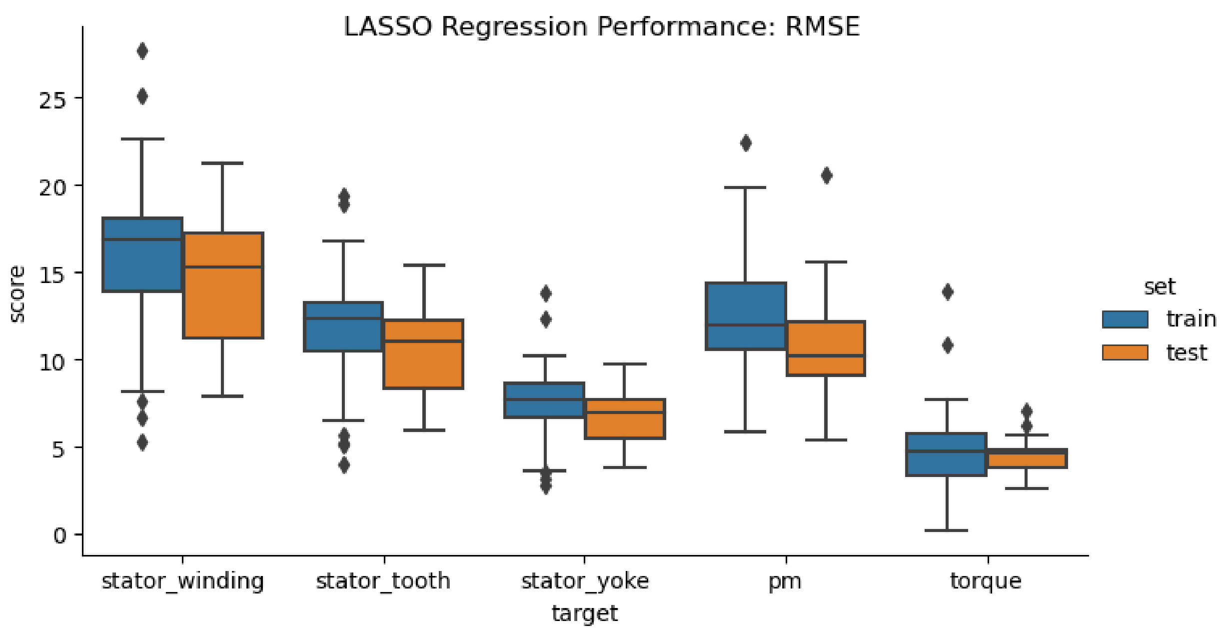

Figure 12: LASSO regression $RMSE$ scoring

LASSO regression looks to perform roughly as well as the Ridge regression in the previous model.

6.5 Polynomial Regression

Polynomial regression works by creating polynomial features and then fitting them onto a linear regression model. As the number of polynomial features quickly multiply as degree increases, a small grid search of degrees from 1 to 4 were tested. The results are shown in Figure 13, below:

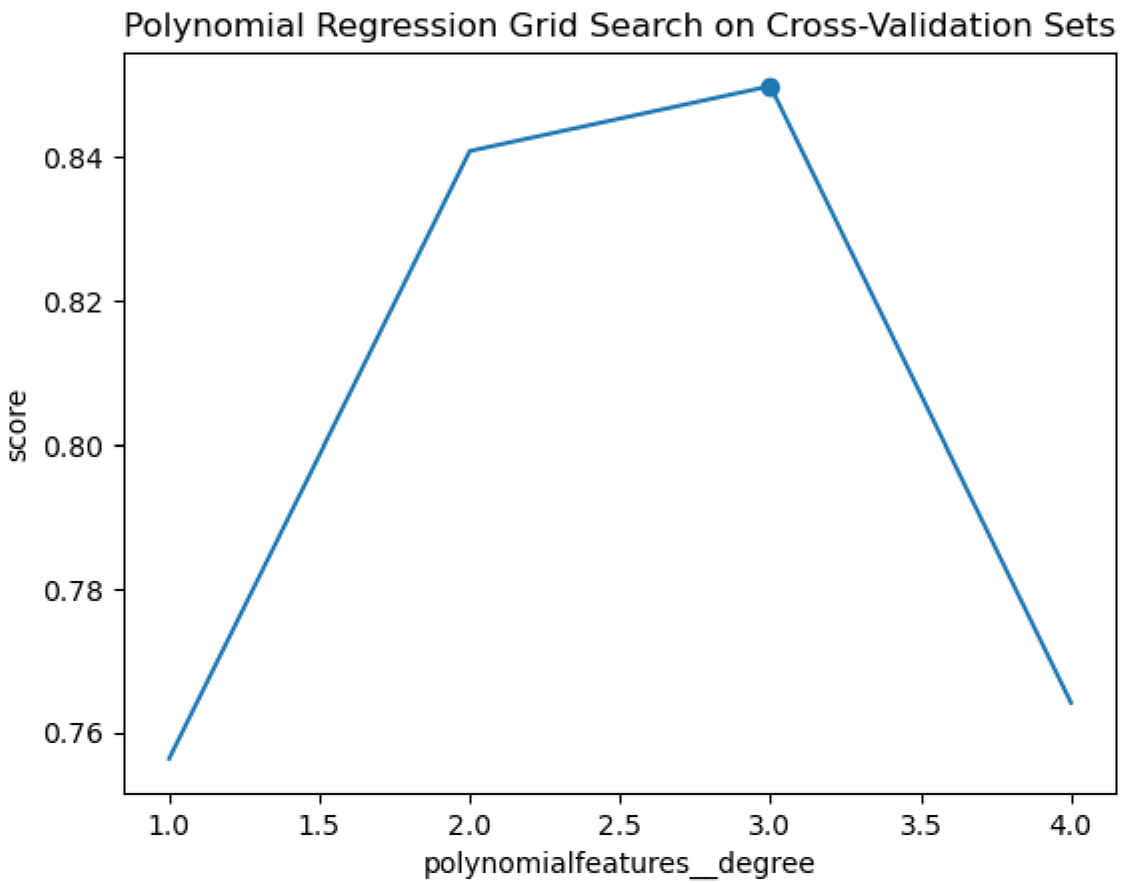

Figure 13: Polynomial regression grid search results on cross-validation sets

The grid search determined that a degree of 3 produced the best model scoring. Using this degree of 3 polynomial dataset, a linear regression model was trained and tested, giving the scoring metrics below, in Figure 14 and Figure 15:

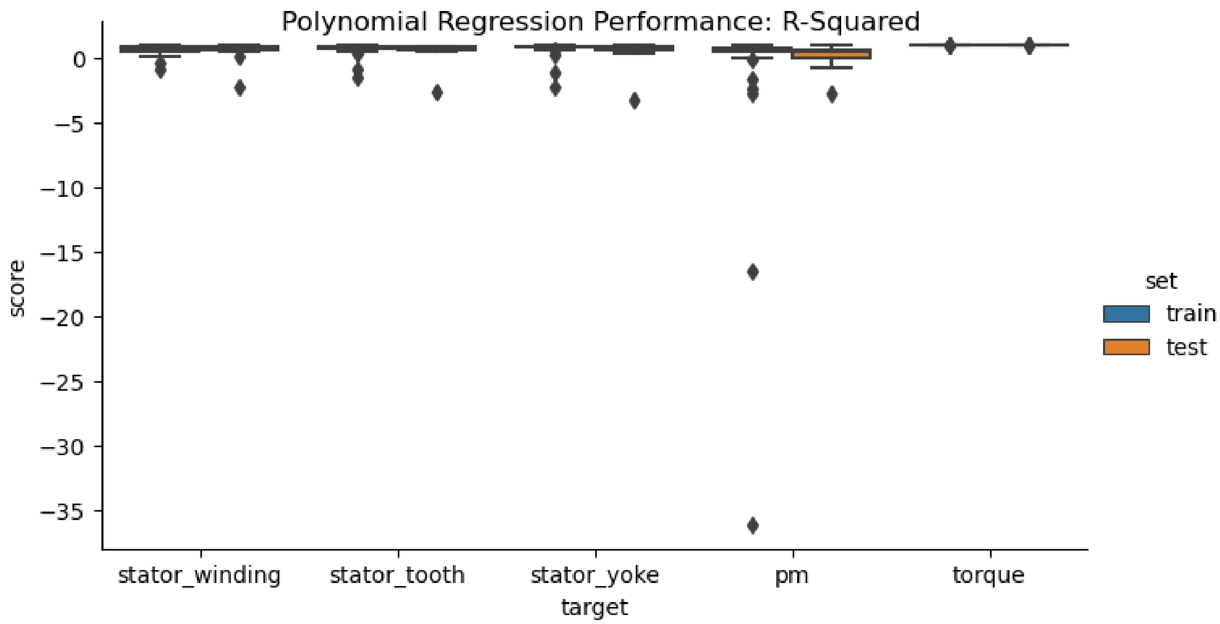

Figure 14: Polynomial regression R^2 scoring

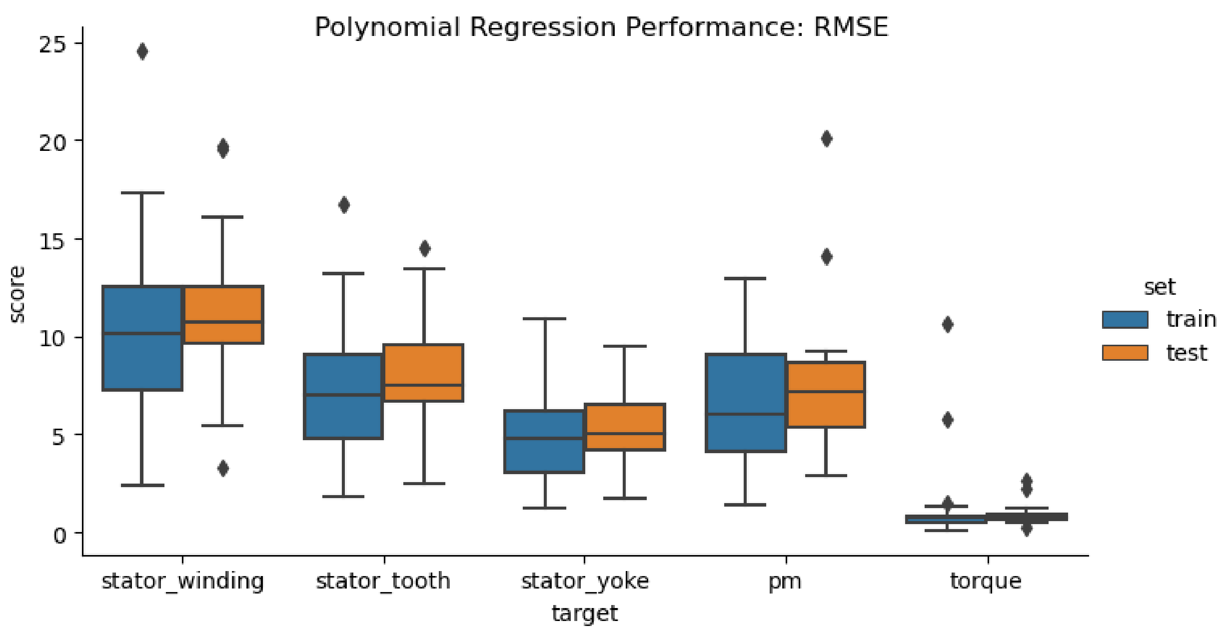

Figure 15: Polynomial regression $RMSE$ scoring

The polynomial model is still producing negative R^2 scores for motor temperature prediction, however, the error in *torque* prediction has been reduced when compared to the previous models.

6.6 Exponentially Weighted Moving Average (EWMA) Regression

Creating EWMA features in a preprocessing step prior to fitting to a linear regression model is the method recommended by the research paper mentioned in the beginning of this report. The previous models had at most one hyperparameter to tune. This EWMA model will have two hyperparameters:

1. How many EWMA features to choose.
2. The spans for each feature.

Since, there is a very high number of possible combinations for the hyperparameters, a repeated random sample for spans for each EWMA feature can be done to get a distribution of results. From this distribution of results, the optimal number of EWMA features can be chosen and then the best combination of spans can be chosen.

Figure 16, below, plots boxplots of 30 random samples of span size for a given number of EWMA features created:

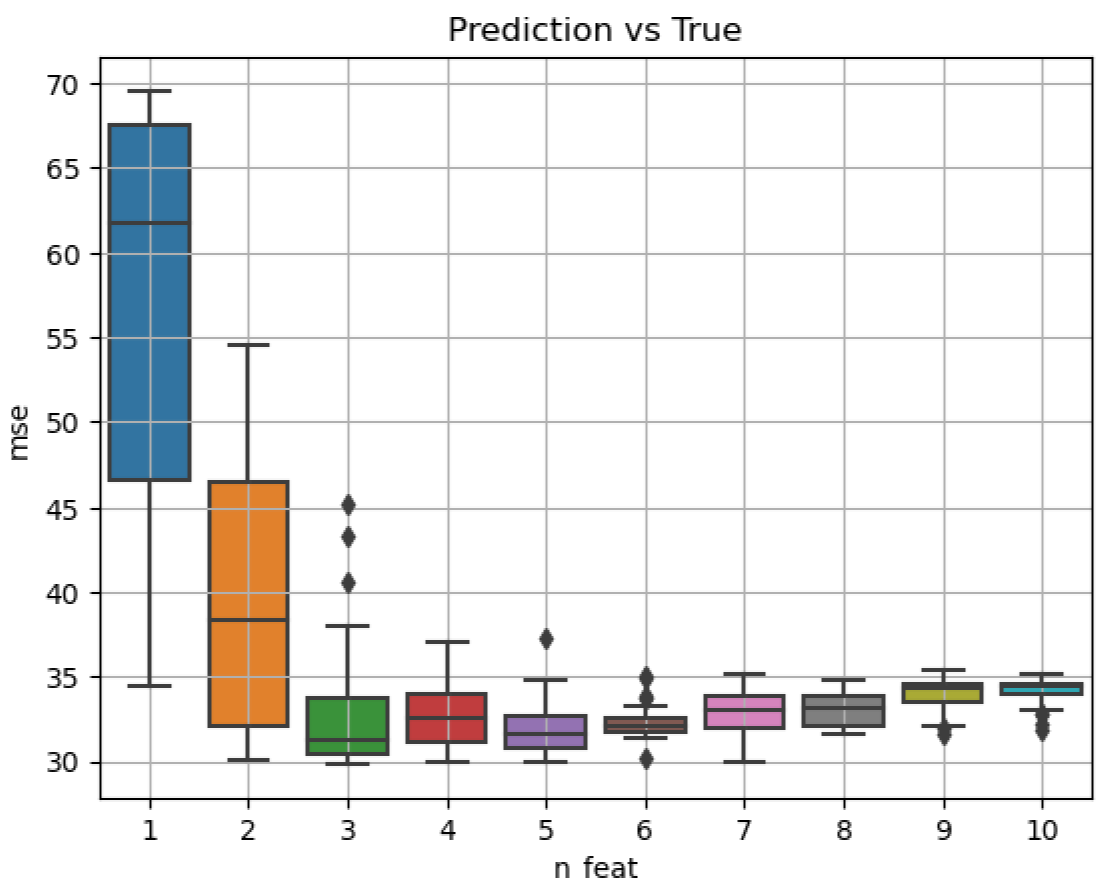

Figure 16: Model mean-squared-error as a function of number of EWMA features for all target variables

The spans ranged from 1 to 60 minutes (when data was available), chosen by a random draw each repetition. It is important to note, that given the nature of predicting current motor temperatures, only past sensor data was used to avoid leakage of “future” data into the model predictions.

The EWMA model appears to be predicting the best when it has 4-5 EWMA features. It is also to be noted that as the number of features increase, the span chosen for each feature has less of an impact on the model's accuracy (the spread of the data decreases with the number of features).

Since torque is already being predicted well, we will continue by focusing on predicting just the motor temperatures. Figure 17, below, shows the same type of plot as Figure 16, however, this time only the motor temperatures are considered in the scoring:

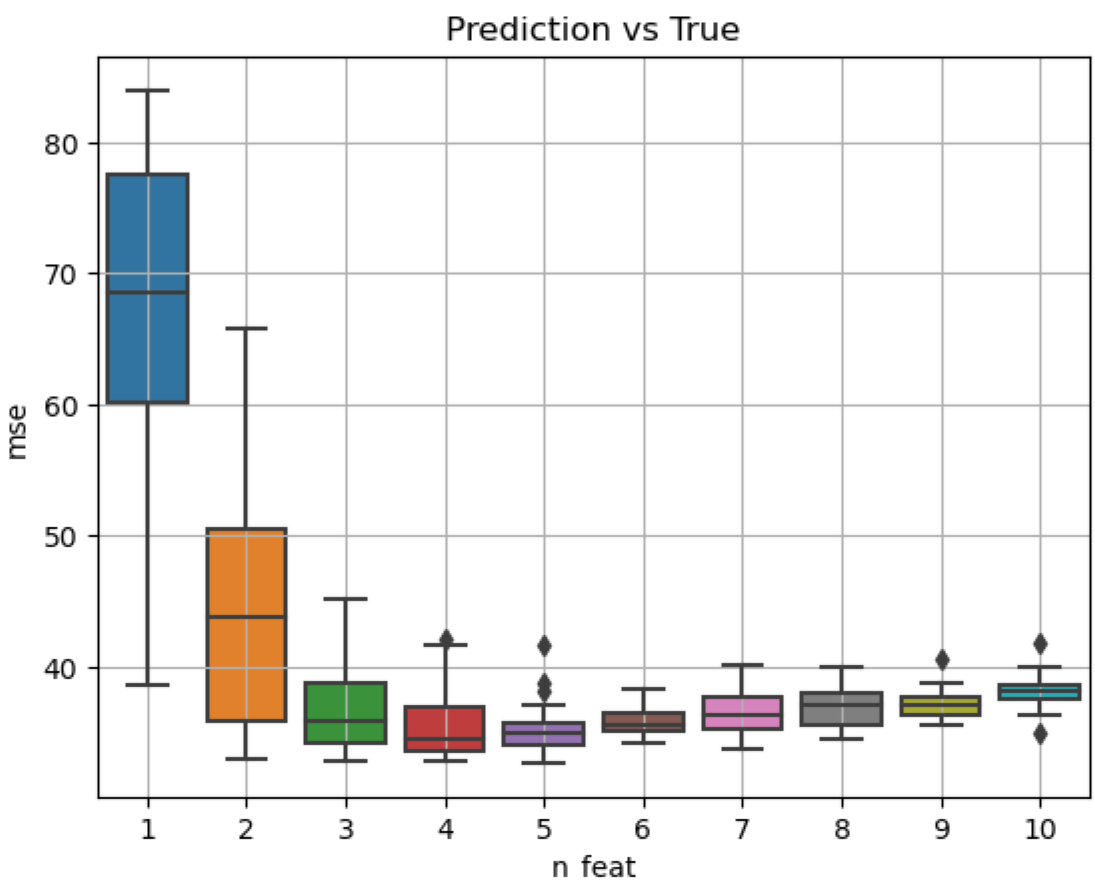

Figure 17: Model mean-squared-error as a function of number of EWMA features for motor temperatures only

When running the tests again but this time with only temperatures, the model scores did get worse, but this was expected due to torque being predicted well in all models so far. The other thing to note is that the optimal number of features appears to be four (by finding the number of features that produce the lowest median score). Now that we know how many features to collect, we need to determine the best span values for the four features.

By collecting all unique combinations of span values, each ranging from 1 to 60 minutes, for 4 EWMA features, and a score for a model trained on each of them, we get the scatterplot below, in Figure 18:

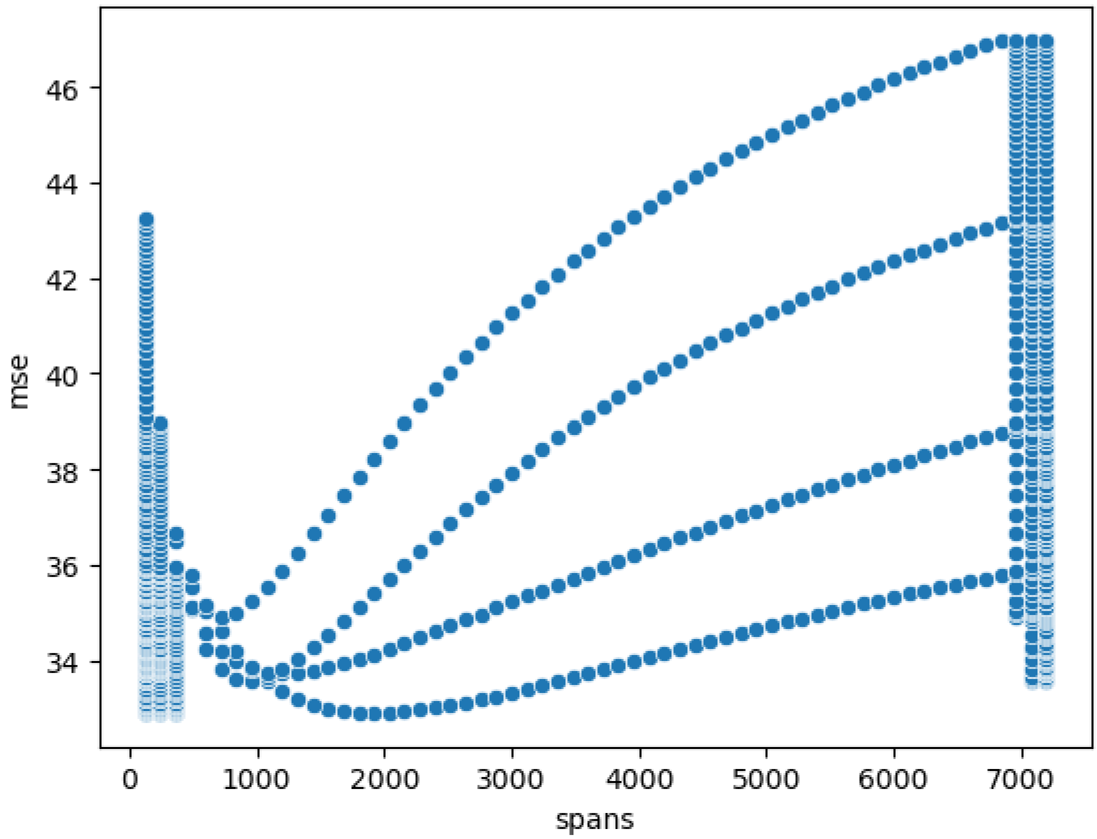

Figure 18: Model score plotted against EWMA feature spans

According to the plot above, the features that have spans closer to 0, meaning the most recent information, tend to score better than those that use old information. This intuitively makes sense. The temperature should be most affected by recent actions in the motor and not the really old actions. The best spans for the four features are 1, 2, 3 and 16 minutes. Let's see how well the target temperatures are predicted using these hyperparameters for the EWMA model. Figure 19 and Figure 20, below, show the EWMA model's scoring metrics:

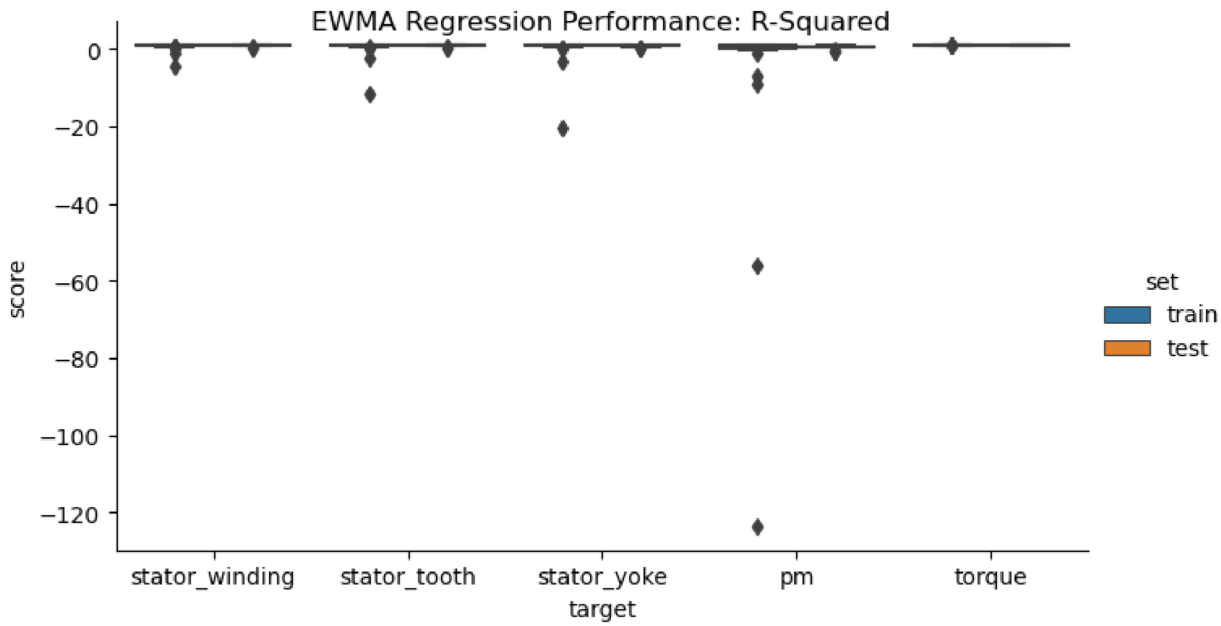

Figure 19: EWMA model R^2 scoring

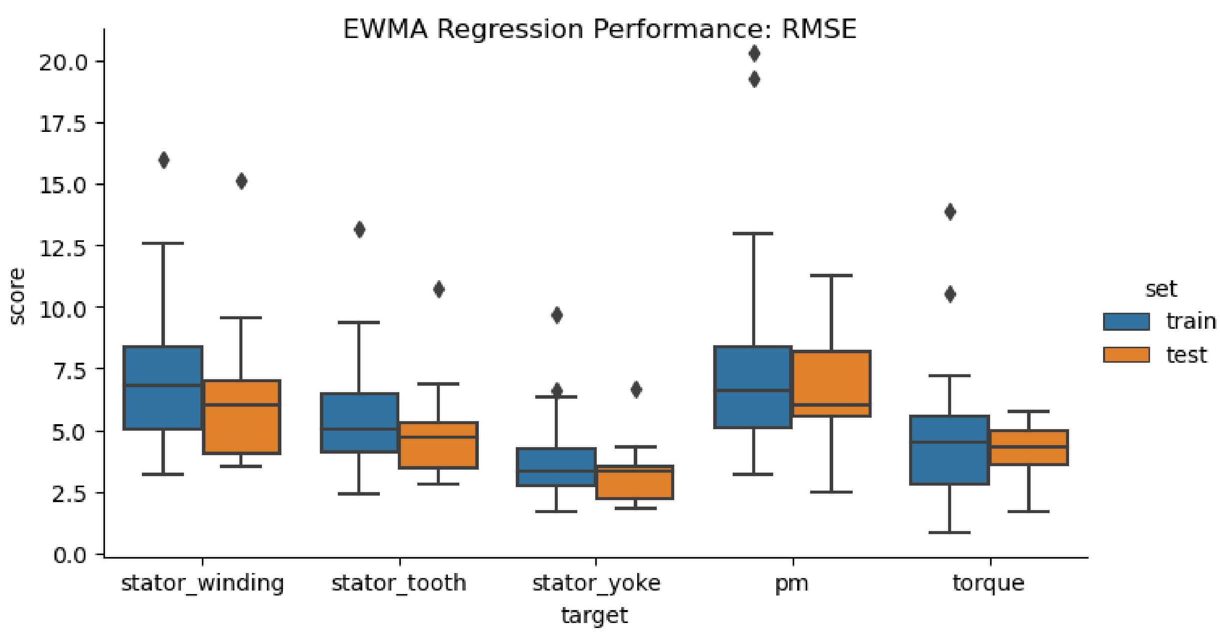

Figure 20: EWMA model $RMSE$ scoring

Although there are still a few outliers that produce negative R^2 scores, the Exponentially Weighted Moving Average regression model definitely outperforms the other models when it comes to temperature prediction. This can be visualized better in Figure 21 and Figure 22, below:

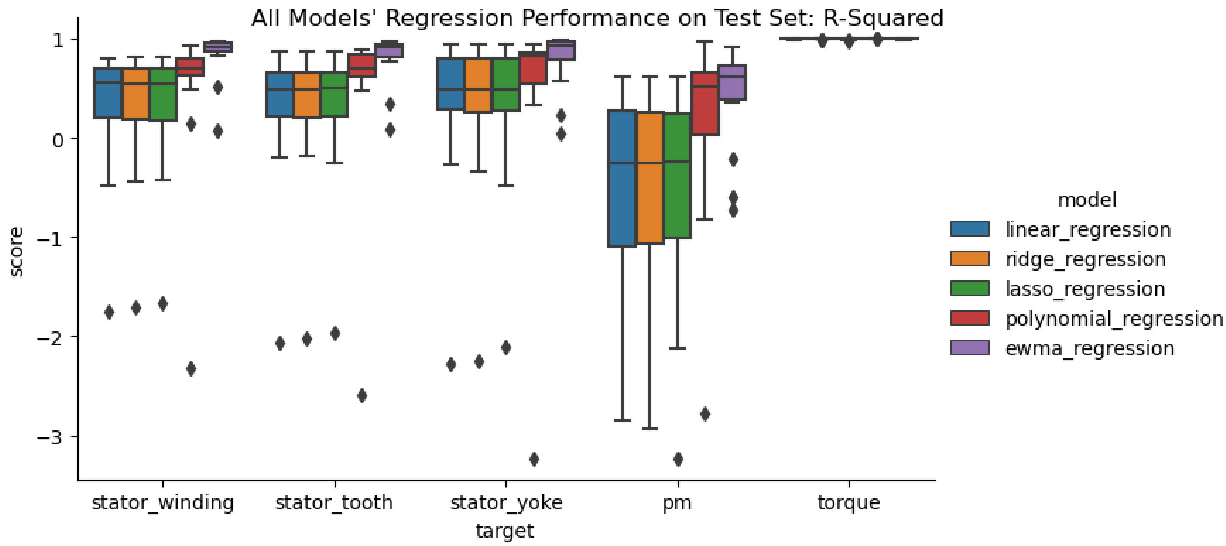

Figure 21: All model R^2 scoring

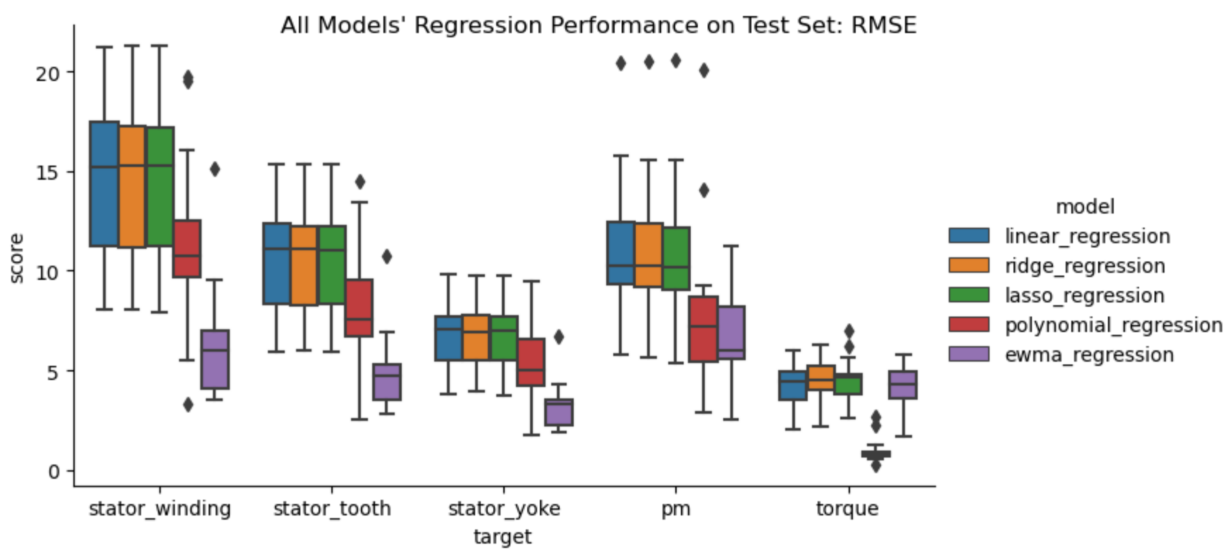

Figure 22: All model $RMSE$ scoring

For all temperature predictions, the EWMA model predictions have the lowest error. For torque prediction, the polynomial model predictions have the lowest error.

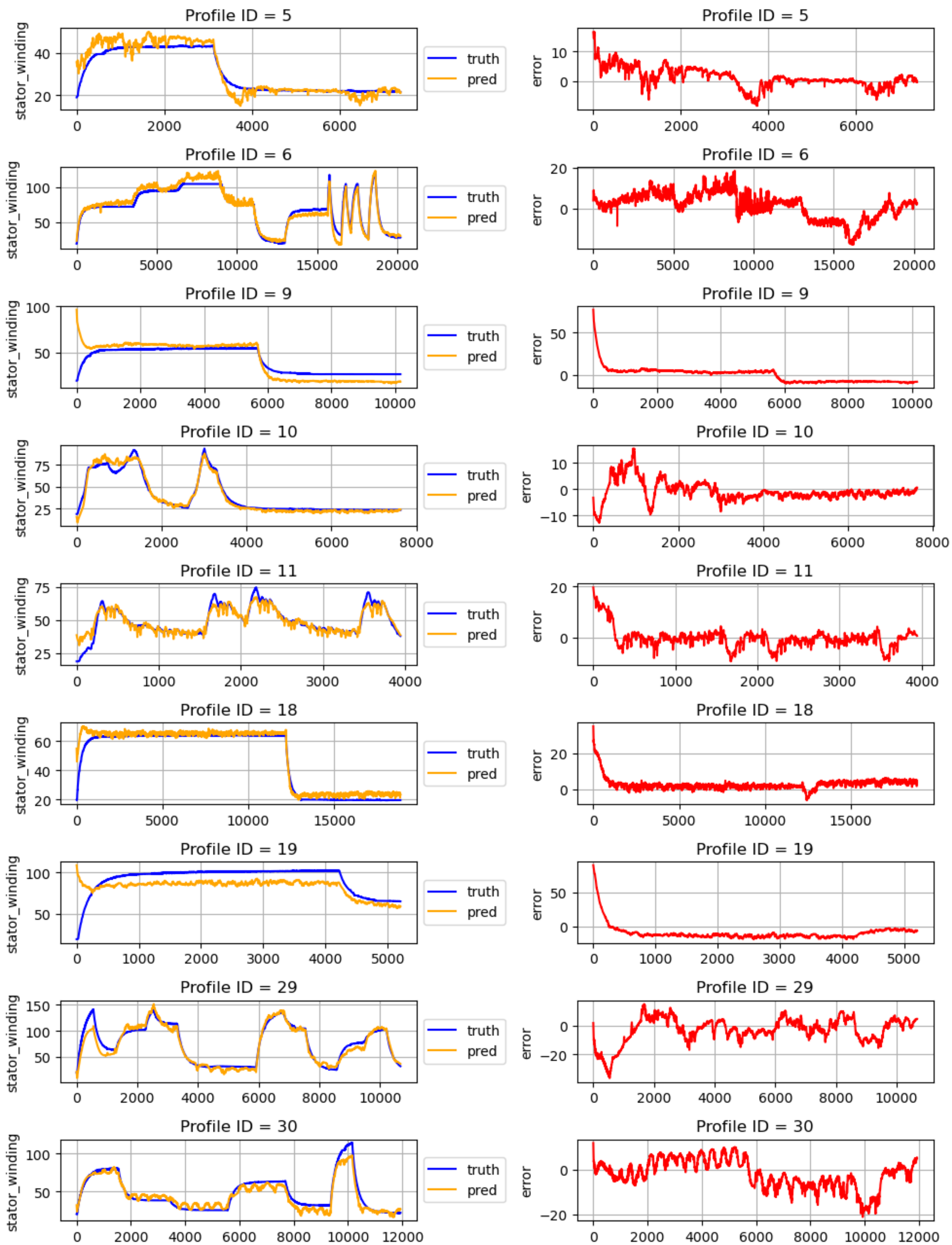

Figure 23a: Stator winding truth vs prediction and error

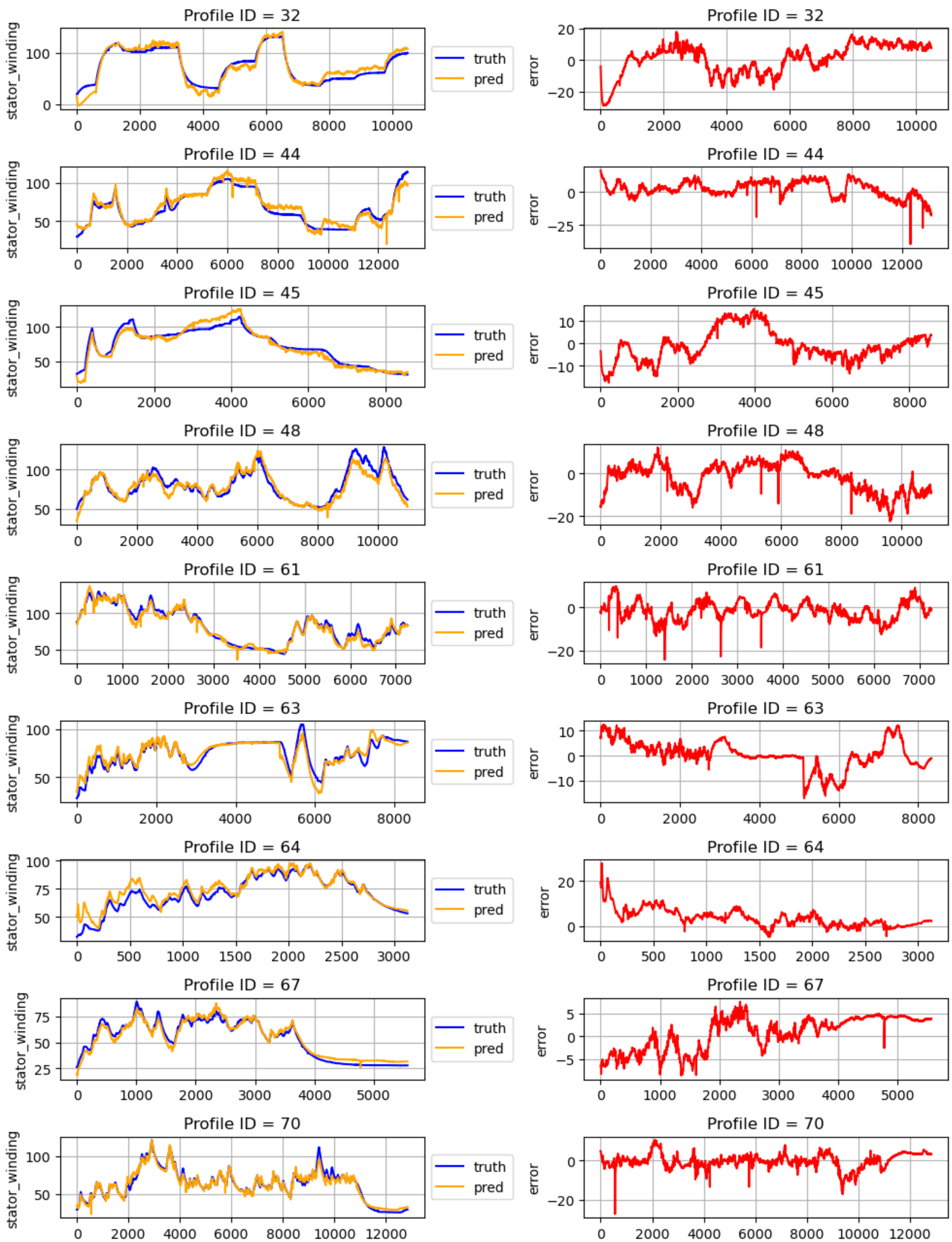

Figure 23b: Stator winding truth vs prediction and error

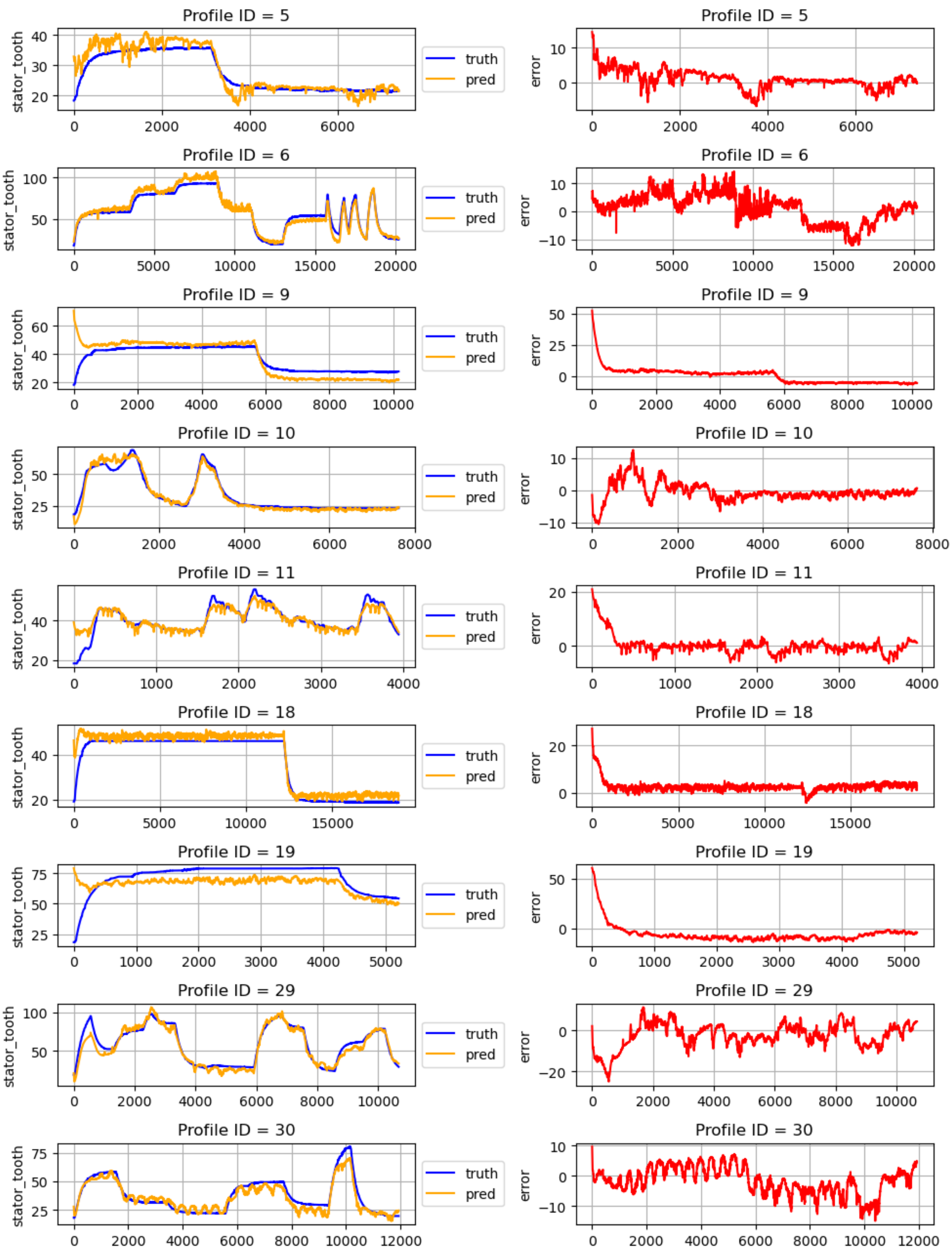

Figure 24a: Stator tooth truth vs prediction and error

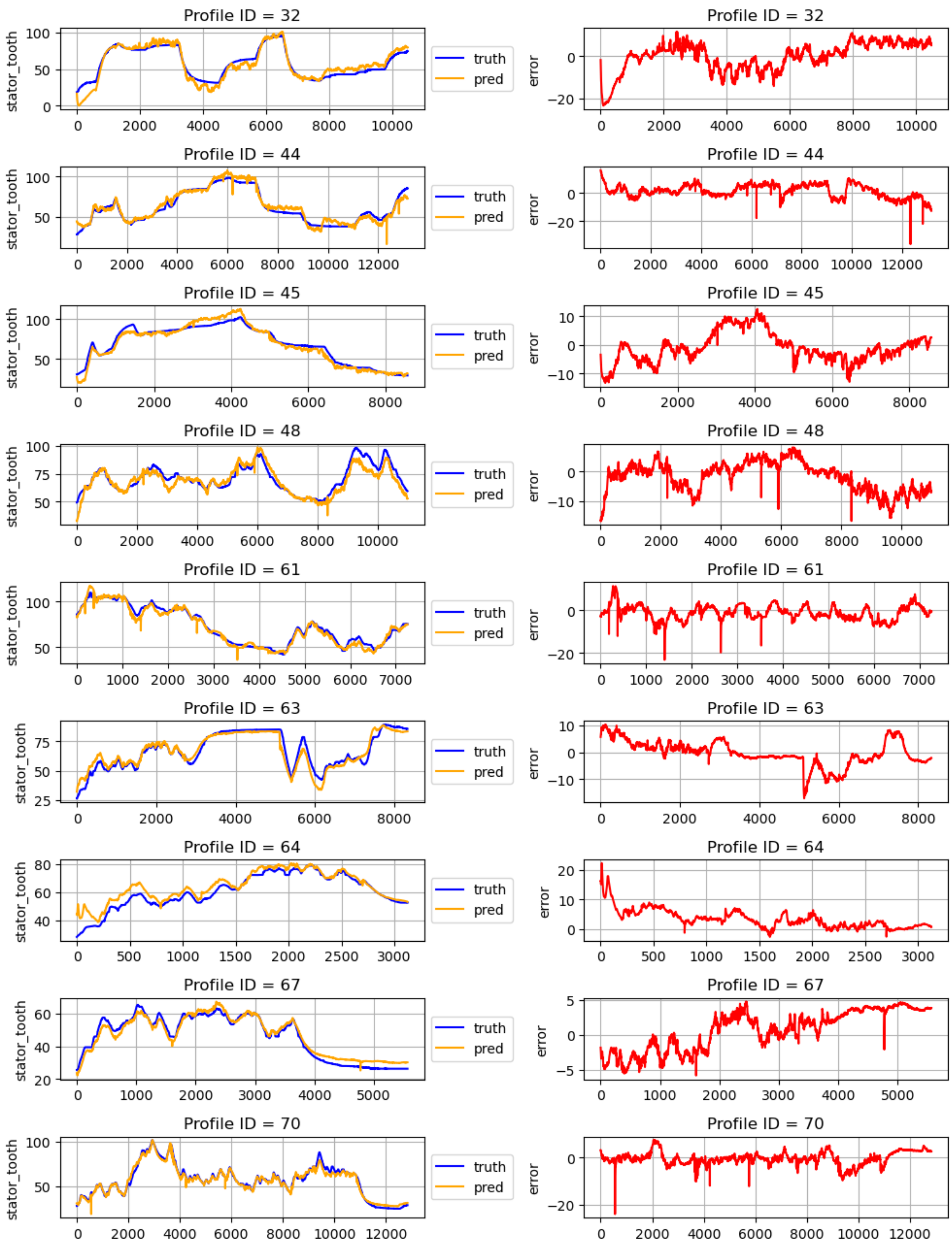

Figure 24b: Stator tooth truth vs prediction and error

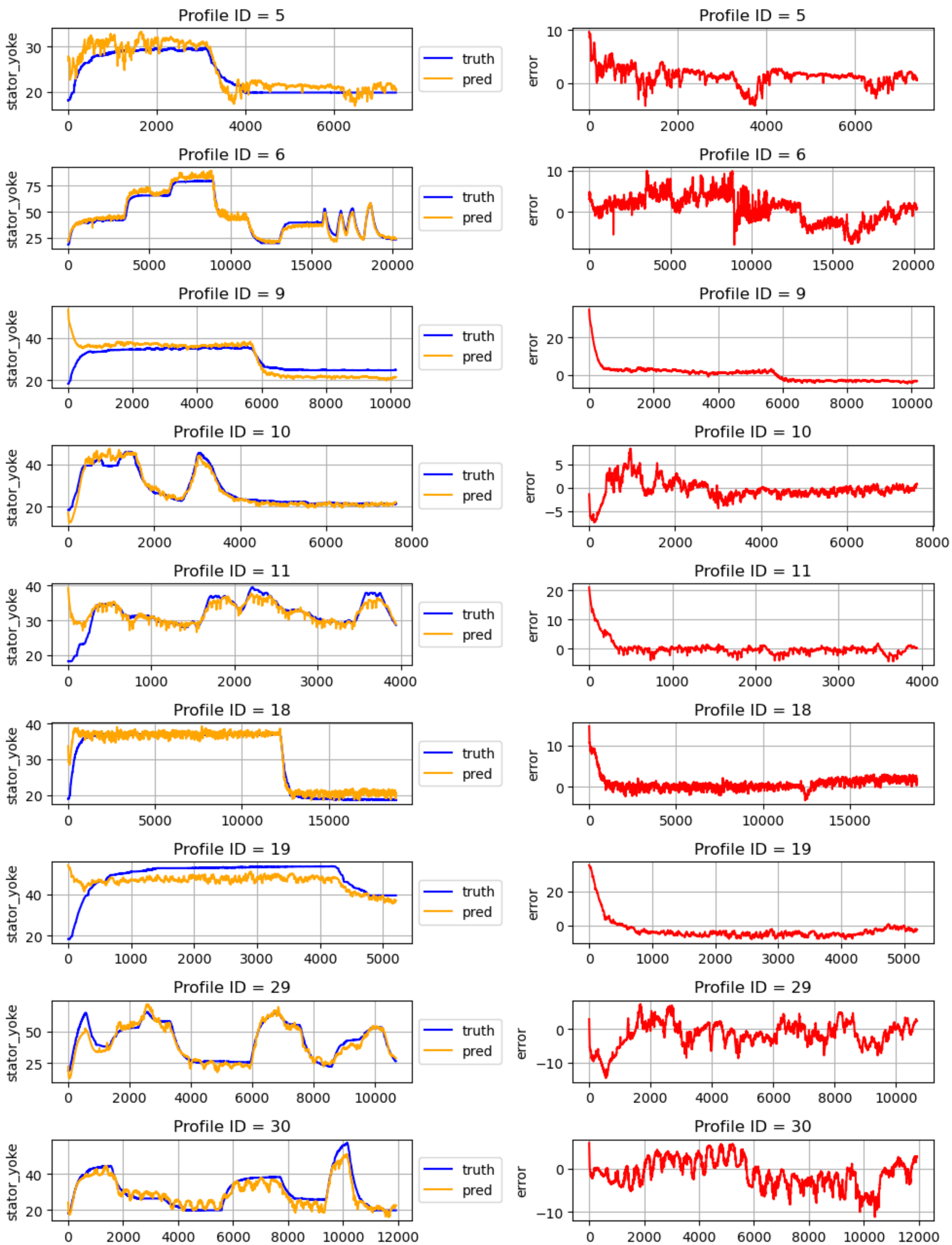

Figure 25a: Stator yoke truth vs prediction and error

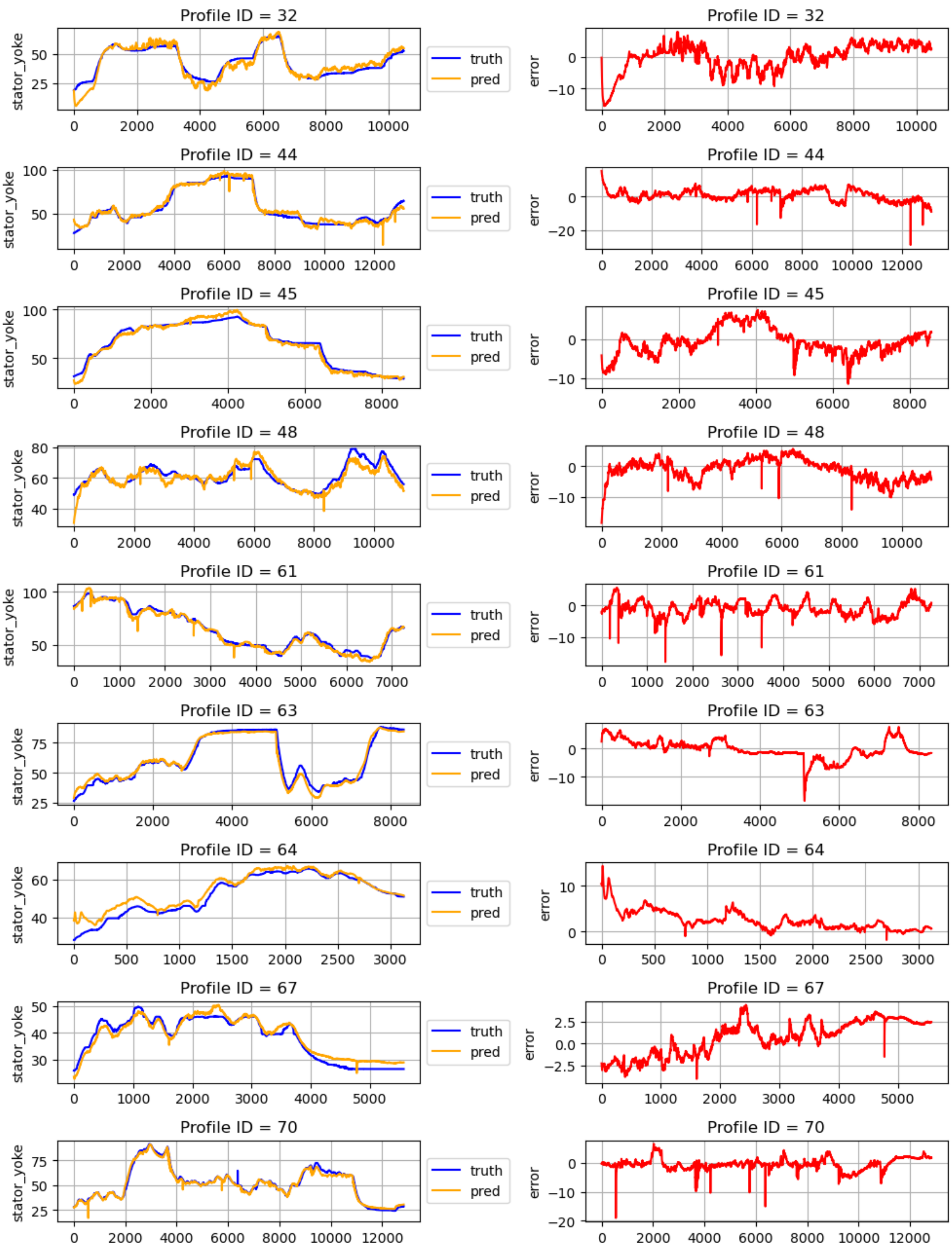

Figure 25b: Stator yoke truth vs prediction and error

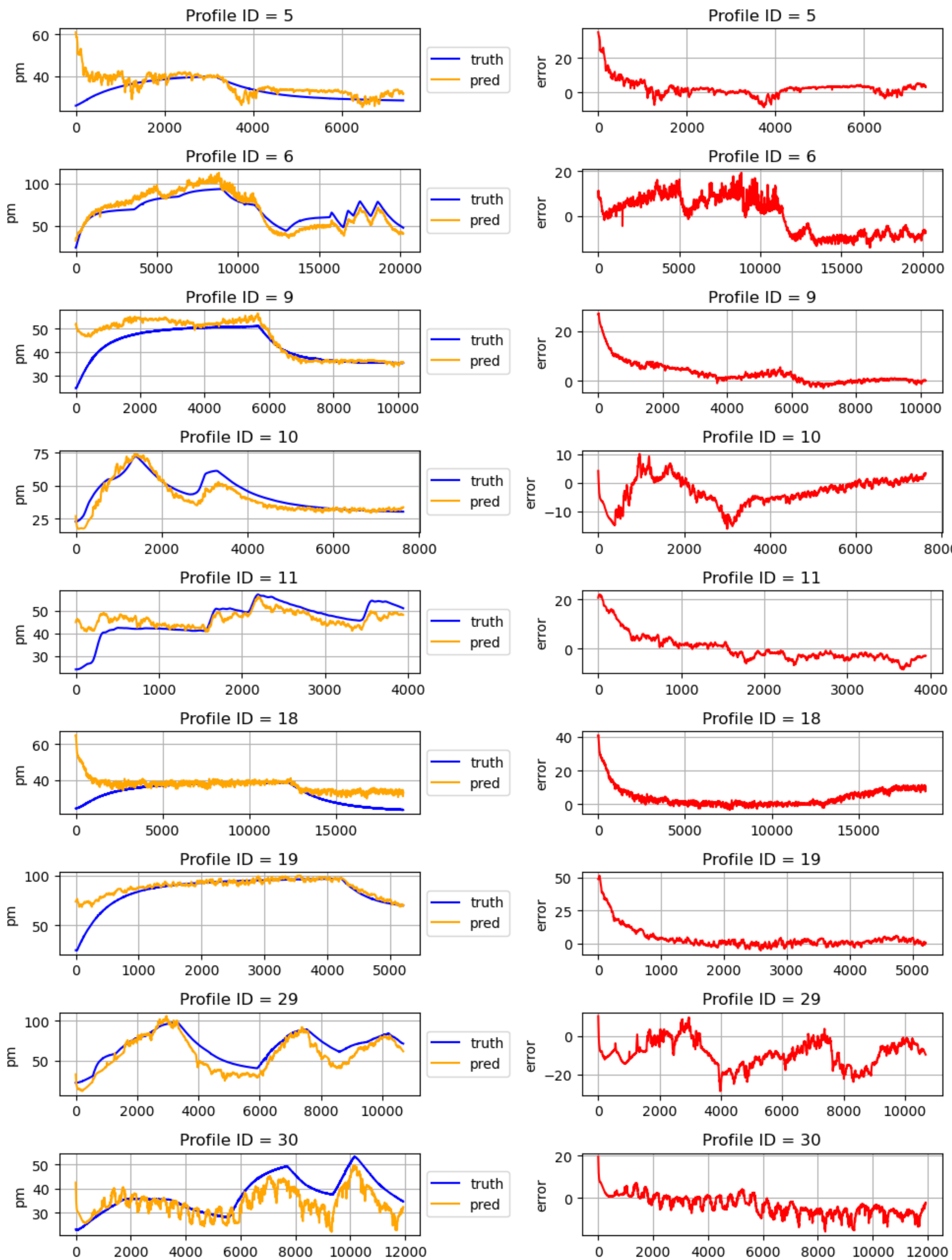

Figure 26a: Pm truth vs prediction and error

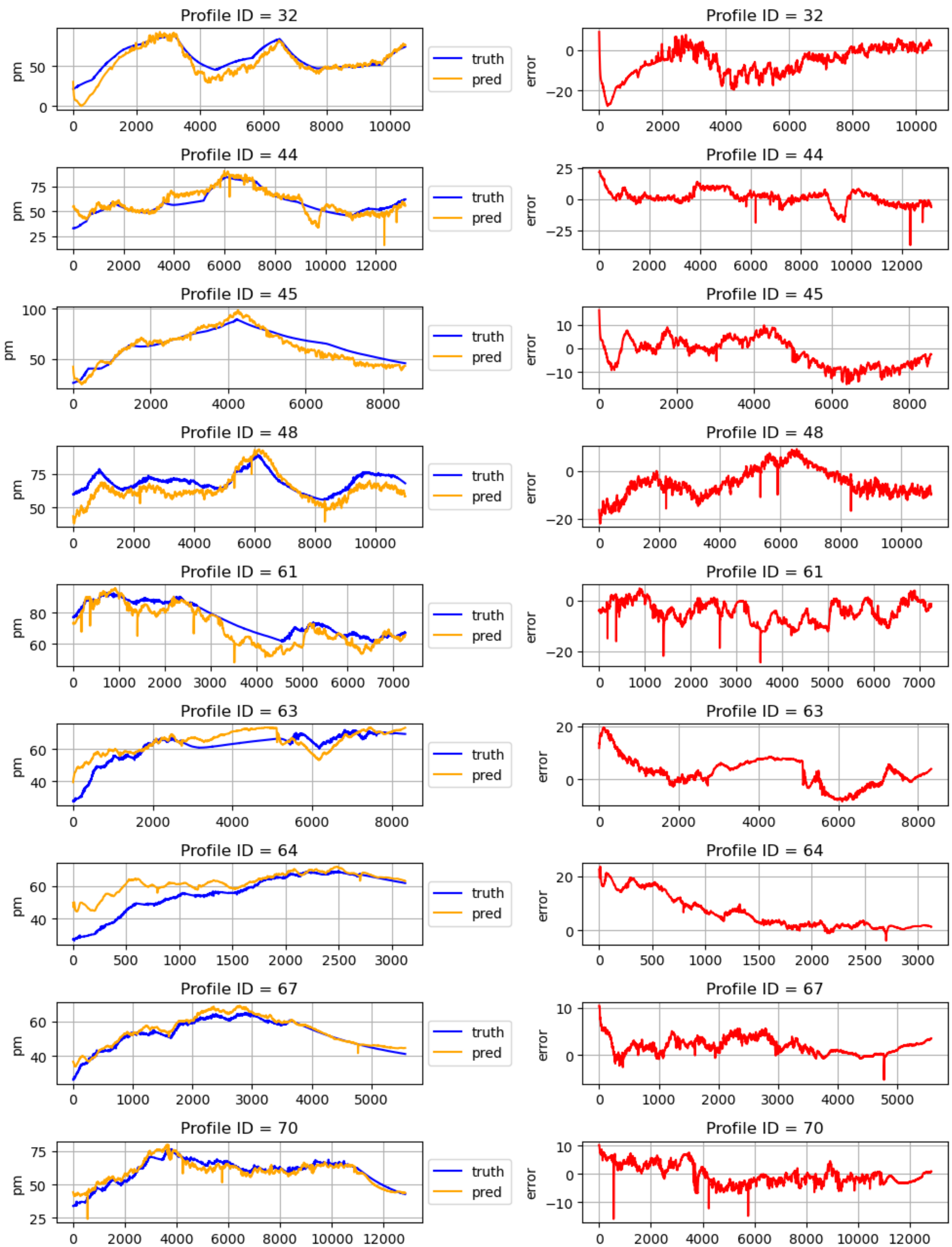

Figure 26b: Pm truth vs prediction and error

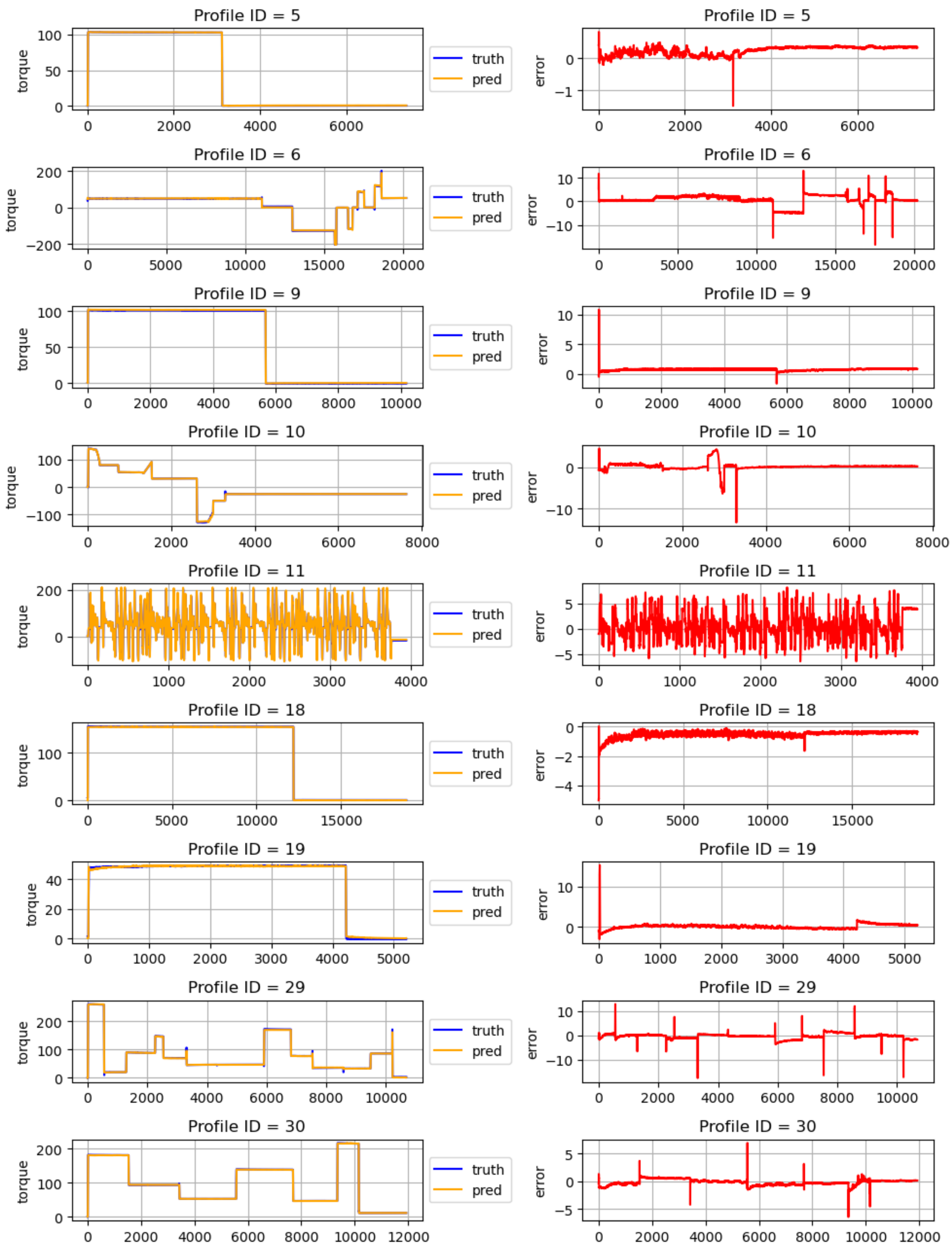

Figure 27a: Torque truth vs prediction and error

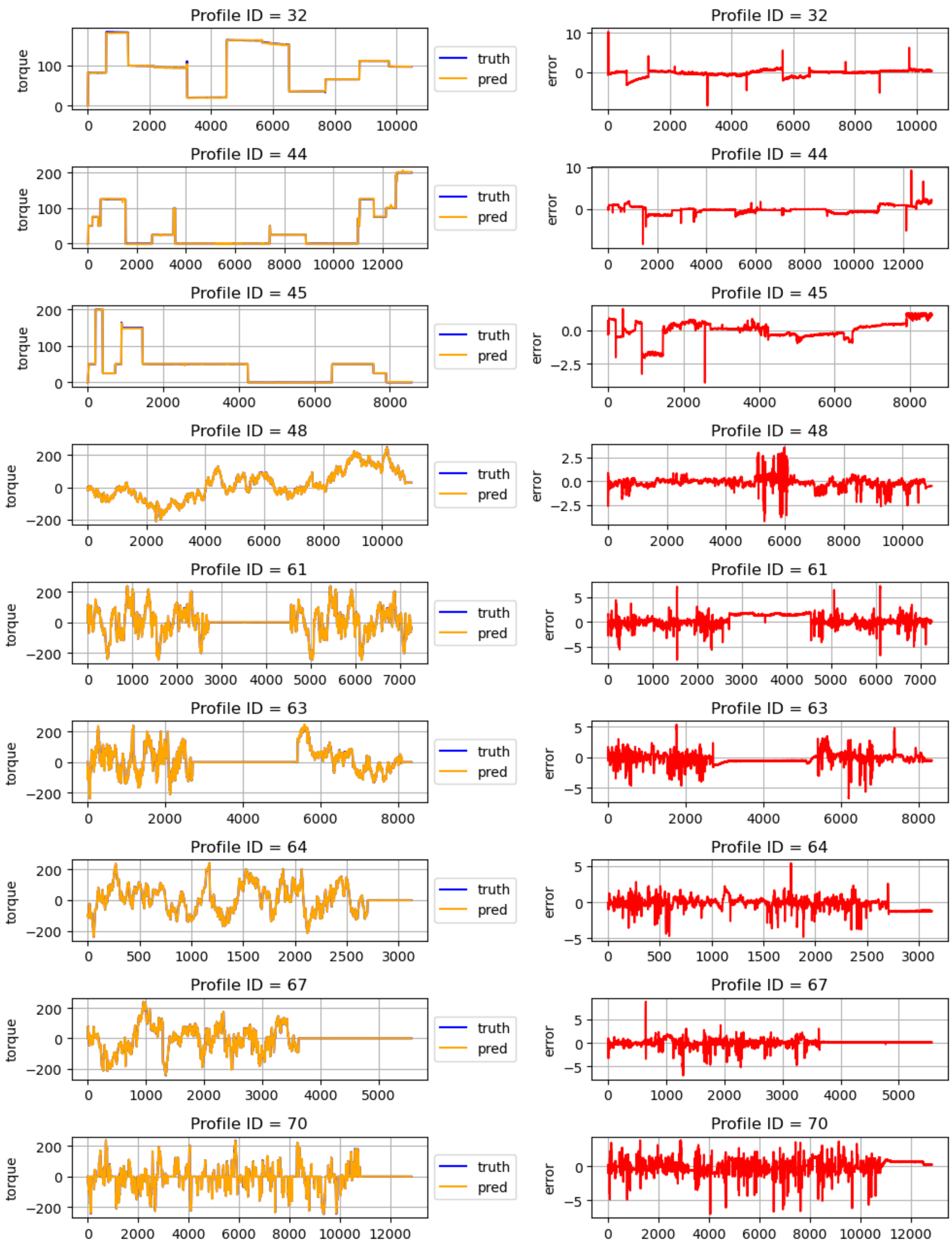

Figure 27b: Torque truth vs prediction and error

7.0 Conclusions and recommendations

The EWMA model and the polynomial model are doing a great job at capturing the trends in the target variables and closely match their magnitudes for almost all points of each measurement session collected during testing. This information strongly suggests that the target variables of the motor could indeed be predicted from measurements.

Although, this is clearly not a perfect solution. There are sometimes high prediction inaccuracies at the beginning of a measurement session due to a history of data not existing for the creation of EWMA features in the temperature model. Future work can include:

1. Exploring methods to lower error spikes at the beginning of each new motor session.
2. Optimizing the predictive models to be fast enough for real time application on a vehicle's computer.

If the motor temperature predictions are accurate enough for a given PMSM manufacturer, then there will be no need to rely on temperature sensors within the system if adequate model training can be done prior to commercial production. This will greatly benefit scenarios where sensors fail and are inaccessible due to the nature of the motor design.

8.0 Bibliography

Britannica, T. Editors of Encyclopaedia (2023, May 18). Ohm's law. Encyclopedia Britannica. https://www.britannica.com/science/Ohms-law

IEA (2022), Electric Vehicles, IEA, Paris https://www.iea.org/reports/electric-vehicles, License: CC BY 4.0

Kirchgässner, Wilhelm & Wallscheid, Oliver & Böcker, Joachim. (2019). Empirical Evaluation of Exponentially Weighted Moving Averages for Simple Linear Thermal Modeling of Permanent Magnet Synchronous Machines. https://www.researchgate.net/publication/331976678_Empirical_Evaluation_of_Exponentially_Weighted_Moving_Averages_for_Simple_Linear_Thermal_Modeling_of_Permanent_Magnet_Synchronous_Machines

Wilhelm Kirchgässner, Oliver Wallscheid, & Joachim Böcker. (2021). Electric Motor Temperature [Data set]. Kaggle. https://doi.org/10.34740/KAGGLE/DSV/2161054