Predictive Calibration of Fluxgate Magnetometer Offsets on the Cluster Spacecraft using Machine Learning

Joey Wee. Imperial College London.

***Abstract*—KNN, SVR, FNN, and LSTM machine learning models are used to predictively calibrate the offsets of the Cluster mission magnetometers from telemetry. Feature analysis reveals that PSU and instrument temperatures influence the offset. Model predictions yielded mean absolute errors of 0.19 ± 0.05 nT (KNN), 0.24 ± 0.07 (SVR), 0.21 ± 0.05 nT (FNN), and 3.44 ± 0.26 nT (LSTM), all exceeding the 0.05 nT error of traditional methods. While the offset is partially characterised by the temperatures, further unknown variables also have an effect. Additional data is required to identify these variables. FNN and LSTM models struggled to converge with available data. It was concluded that machine learning is not well suited to predictive calibration applications.**

I. Introduction

The four Cluster spacecraft in Earth orbit each carry a three-axis fluxgate magnetometer (FGM). Each axis has a zero error (offset) which must be calibrated. The spacecraft rotate about their X axes at 0.25 Hz, causing the spinning Y and Z axes to exhibit signals at the rotation frequency if the offset is not calibrated. These offsets are calibrated by minimising the signal at the rotation frequency. [1] However, as the X axis is non-rotating, it cannot be calibrated in this manner. Instead, these offsets must be determined when the spacecraft enters the solar wind for a few months each year. Offsets in between these periods must be interpolated.

The Cluster satellites also collect telemetry about themselves. By determining which underlying variables affect the offset and how, the offsets could be derived from telemetry. Since human input is presently required for the offset calculations, this could simplify the FGM instrument calibration. Relating the offsets to the telemetry would also allow for better methods of estimating the X axis offsets than by linear interpolation.

Because which variables influence the offset and how are unknown, it is not possible to simply fit a function to this data. However, with over 3000 orbits of data per satellite since August 2000, the volume of labelled data might make it suitable to use a supervised machine learning approach to find trends in the offset-telemetry data. To test the applicability of using these techniques for predicting the offsets, four types of ML models will be trained and tested on the data: K-nearest neighbours (KNN), support vector regression (SVR), feedforward neural network (FNN), and long short term memory (LSTM) models.

TABLE I: Codes and Names of Shortlisted Telemetry

Spacecraft Telemetry Code	Telemetry Name
F034	-12 Volt Line
F047	+5 Volt Line
F048	+12 Volt Line
F055	PSU Temperature
F074	FGM Temperature
J106	Main Bus Voltage
J213	Main Bus Current
J216	Solar Array Current
J236	Supply Current

II. Data

The Cluster FGMs have multiple sensitivity ranges but spend most of the time in range 2. As such, analysis will be performed using range 2 data only. As the offset is only determined per orbit, telemetry data while in range 2 was averaged over the orbit for feature vectors.

Errors from faulty readings, demarcated by zeroes in the data, were filtered. To account for true zero values, only sudden jumps to zero were discarded. Orbits including eclipses by the Moon or Earth were also removed as these events skew the telemetry averages.

Since including irrelevant telemetry as features would introduce noise into the predictions, the telemetry streams meaningfully affecting the offset had to be identified. A longlist of telemetry streams that might influence the offset was selected (Table I). Potential correlations between telemetry and the offset can be verified by plotting telemetry averages against offsets (Figs. 1 and 2) and telemetry averages with offsets against time (Figs. 3 and 4).

Fig. 1 plots offsets against F074 (FGM Temperature) telemetry, showing a clear negative correlation between offset and temperature. The correlation is also visible in the variations of the F074 telemetry and offsets in time (Fig. 3): temperature peaks match offset troughs. This implies that F074 is a meaningful input for predicting offsets.

Conversely, the offset varies considerably even when the F034 (-12 volt line) telemetry remains mostly constant around -13.2 volts (Fig. 3). The voltage-independence is also apparent when plotting F034 and offset against time (Fig. 4), where they do not vary with or against each other, suggesting that F034

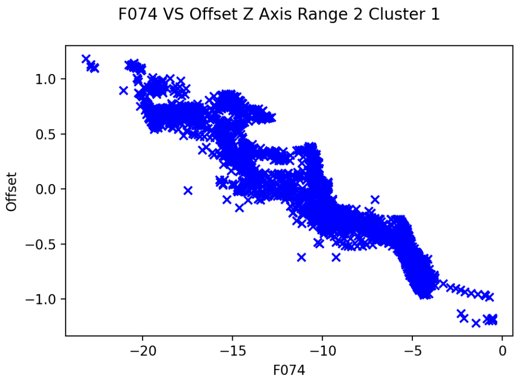

Fig. 1: The effect of variation in the F074 telemetry against the offset.

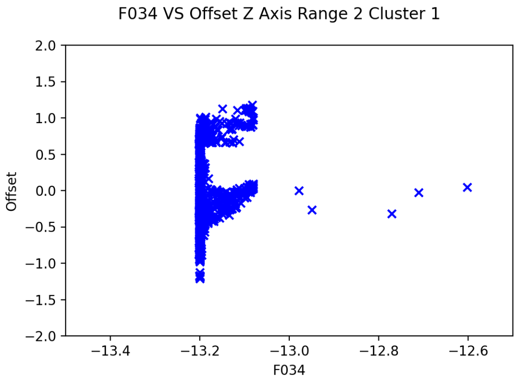

Fig. 2: The effect of variation in the F034 telemetry against the offset.

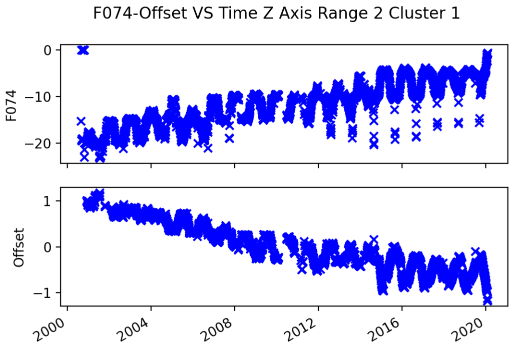

Fig. 3: The variations of the offset and the F074 telemetry in time.

is not a meaningful feature as the offsets mostly vary by other variables.

Feature meaningfulness was also determined by training a support vector regression (SVR) model to predict the offsets based on one telemetry feature and comparing the predictions to reference offsets. Fig. 5 shows the predictions of a model trained only on the F074 telemetry; it shows that the F074 telemetry is meaningful as the predictions based on it follow the offsets well. However, for the F034 telemetry, a similar plot (Fig. 6) shows that it does not carry enough information for the model to predict the telemetry at all, proving that F034 is not meaningful.

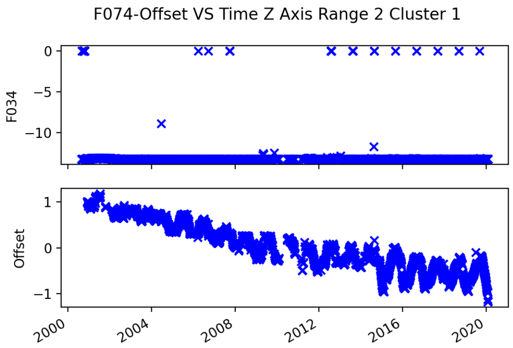

Fig. 4: The variations of the offset and the F034 telemetry in time.

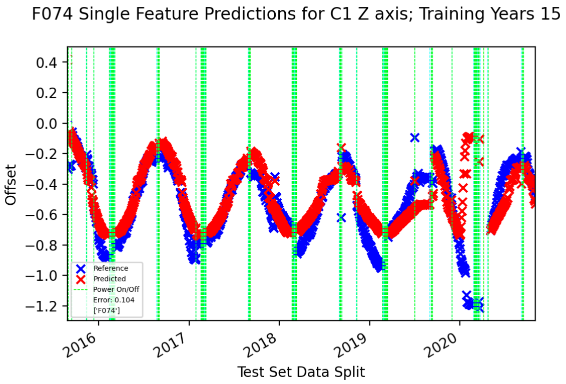

Fig. 5: The predicted offsets of an SVR model trained only on the F074 feature against the reference offsets. The model was trained on the first 15 years of Cluster 1's Z axis data and the remaining data was plotted.

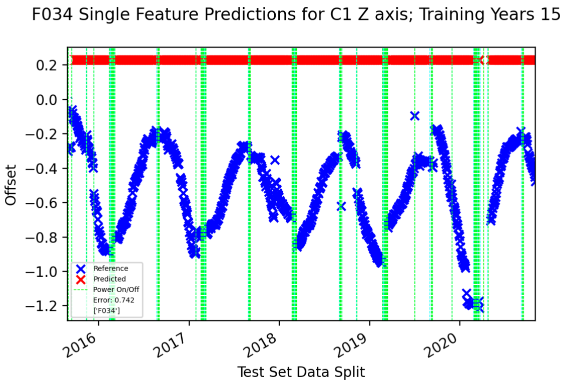

Fig. 6: The predicted offsets of an SVR model trained only on the F034 feature against the reference offsets. The model was trained on the first 15 years of Cluster 1's Z axis data and the remaining data was plotted.

Performing this analysis on all telemetry streams identified only F074 and F055 as useful indicators of the offset.

TABLE II: Mean Absolute Errors for KNN Models

Cluster	Y	Z
1	0.57 ±0.26	0.09 ±0.06
2	0.09 ±0.07	0.17 ±0.11
3	0.13 ±0.10	0.15 ±0.11
4	0.18 ±0.11	0.09 ±0.06

III. Training and Testing Datasets

The dataset was split chronologically using a split time: orbits before the split time formed the training set and orbits after formed the testing set. By specifying the number of years of training data to use, the split time was calculated from the time of the first orbit. This reflects the application where the models are trained on historical data and applied to future data.

IV. Model Training and Results

To compare model accuracy, the four regression models were trained on 15 out of 20 years of Y and Z axis data (the axes with the most non-interpolated data) from all four spacecraft. Across the 8 rotating axes, this provides for a consistent basis on which to compare the accuracy of the models.

A. *K-Nearest Neighbours*

KNN models synthesise predictions by averaging the labels of the K nearest training data points to the input feature in feature space, where K is a a hyperparameter defining how many nearest neighbours are considered. The KNN models were implemented using the SciKit Learn Python library. Models were trained for K values of 1 through 10 for each of the 8 rotating axis datasets with the averages weighted by distance between the input and training vectors. The model with the lowest mean absolute error (MAE) on the testing data for each axis was selected. Table II shows the MAE for each axis and the range within which one standard deviation of the individual orbit errors lay in nT.

The mean MAE of the KNN models is 0.19 ± 0.05 nT, less accurate than the 0.05 nT accuracy yielded by current methods. It is important, however, to note that the range of the calibrated offsets for each axis vary significantly. So, a more complete understanding of the sources of error can be gained by plotting the predictions and offsets in time.

The predictions of a KNN model trained on Cluster 4 Z axis data (Fig. 7) immediately after the split time line up well with the actual offsets: the frequency and amplitude of the offset's oscillations are matched well by the predictions. However, around 2018, the calibrated offset fluctuations become less consistent, with some variations much smaller then others. The model fails to handle these smaller variations which implies that this behaviour cannot be characterised by the temperature features alone.

When a similar KNN model is applied to the Y axis of Cluster 1, the predicted offsets follow the oscillations of the calibrated offsets correctly (Fig. 8). However, the model fails to predict the magnitude of the offsets correctly. This discrepancy arises from the continual upwards drift of the offsets for this axis, causing the range of offsets in the testing period to fall outside the range of the training data. The KNN model is not able to extrapolate beyond the range of its training data because it relies on averaging past data to make every prediction; this limitation is apparent in how the peaks of the oscillation of the predictions do not exceed the highest offsets in the training data. To address this, a support vector regression model, capable of fitting regressions to data, was explored in the hope that the regression could be extended slightly beyond the training range to predict new offsets.

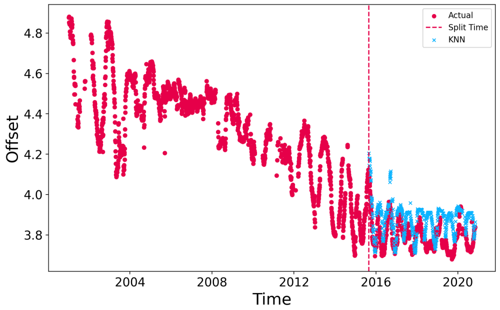

Fig. 7: The calibrated offsets of the Z axis of Cluster 4 (red) with the predicted offsets by a KNN model trained on 15 years of data (blue). The data before the split time is the training data.

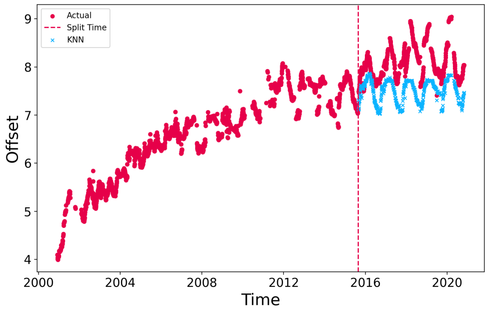

Fig. 8: The calibrated offsets of the Y axis of Cluster 1 (red) with the predicted offsets by a KNN model trained on 15 years of data (blue). The data before the split time is the training data.

V. Support Vector Regression

An SVR model maps data into higher-dimensional space and optimises a hyperplane to fit as many training data points between two support vectors parallel to the hyperplane as possible. The hyperplane shape is controlled by a kernel which allows it to capture non-linear relationships. A radial basis function (RBF) kernel was chosen for this application as it does not require knowledge of how many degrees of freedom are required to characterise the behaviour of the offset.

SVR models have two hyperparameters: γ, which was set to scale automatically based on the input, and c, which controls the strictness of the model with fitting all data points between

TABLE III: Mean Absolute Errors for SVR Models

Cluster	Y	Z
1	0.82 ±0.44	0.23 ±0.26
2	0.11 ±0.09	0.19 ±0.10
3	0.11 ±0.08	0.14 ±0.10
4	0.14 ±0.09	0.18 ±0.11

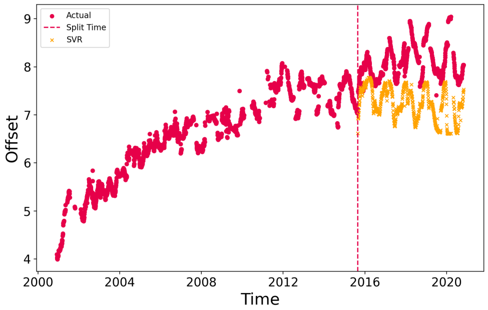

Fig. 9: The calibrated offsets of the Y axis of Cluster 1 (red) with the predicted offsets by an SVR model trained on 15 years of data (orange). The data before the split time is the training data.

the two support vectors. To optimise c, a SciKit Learn SVR model was trained for every c value between 0.1 and 1 in increments of 0.1, selecting the model with the lowest MAE for each axis' testing dataset. As the MAE was evaluated on unseen data, this avoids overfitting and underfitting. The MAEs of the models optimised for each of the 8 spinning axes are shown in Table III.

The mean MAE for the SVR models is 0.24 ± 0.07 nT, higher than the 0.19 ± 0.05 nT of the KNN models and the 0.05 nT of the manual calibration method. While most individual axis errors in Table III are comparable to those of the KNN models, three axes, Cluster 1 Y and Z axes, and Cluster 4 Z axis, show significantly higher errors for the SVR models.

The MAE for the SVR predictions for Cluster 1 are noticeably larger than those for the other spacecraft because Cluster 1's offsets vary over a much larger range than the offsets of the other satellites. In general, the behaviour of the offset and model are similar for all rotating-axis datasets. Hence, the MAE scales proportionally with the range of the offset's variation. This larger variation may stem from Cluster 1's origin as a flight spare, built separately from the other three spacecraft after an initial launch failure. As a result, the mean MAE across the rotating axes serves only as a general performance indicator, rather than an objective metric for model selection.

When the SVR predicted offsets are compared to the calibrated offsets for Cluster 1's Y axis (Fig. 9), it can be seen that, as for the KNN models, the model seems to be able to predict the offset's oscillations but fails to predict offsets outside the range of the training offsets.

This can also be seen when plotting the SVR predictions and calibrated offsets for the Y axis of Cluster 4 (Fig. 10). The training data for this axis is very noisy: the offsets often have very large jumps between values. It has been previously determined that these jumps happen after the instrument has been restarted. These jumps appear unrelated to temperature data or are beyond the model's ability to characterise. However, even for this noisy data, the SVR model is able to predict the general pattern of the oscillation. Despite this, the model still struggles with amplitude prediction and discontinuities. A more complex model, like a feedforward neural network, might better characterise the offset's behaviour.

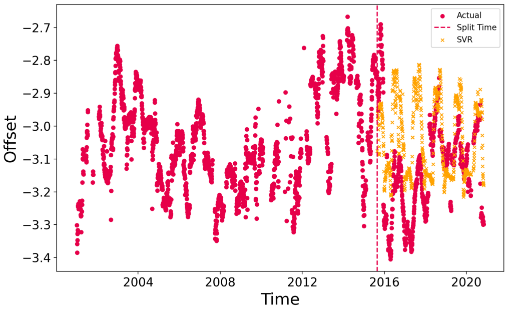

Fig. 10: The calibrated offsets of the Y axis of Cluster 1 (red) with the predicted offsets by an SVR model trained on 15 years of data (orange). The data before the split time is the training data.

TABLE IV: Mean Absolute Errors for FNN Models

Cluster	Y	Z
1	0.42 ±0.29	0.12 ±0.10
2	0.15 ±0.12	0.15 ±0.11
3	0.13 ±0.08	0.13 ±0.09
4	0.12 ±0.09	0.43 ±0.07

VI. Feedforward Neural Network

In an FNN, each telemetry feature is fed into a neuron in the first layer of the neural network and, after passing through multiple layers of neurons with weights and biases, the output neuron returns a prediction. By combining linear neurons, the FNN can approximate complex relationships.

The models were created using the PyTorch library and trained for 100 epochs with the Adam optimiser using the MAE as a a metric. To improve model performance, the features and offsets were scaled using SciKit Learn's RobustScaler. The chosen architecture was defined to include rectified linear unit (ReLU) activation functions to introduce non-linearity to allow the model to fit more complex, non-linear functions:

- Input Layer: 2 features
- Dense Layer (20 units, ReLU)
- Dense Layer (10 units, ReLU)
- Dense Layer (5 units, ReLU)
- Output Layer: 1 unit (no activation function)

The resulting MAEs for each axis are shown in table IV.

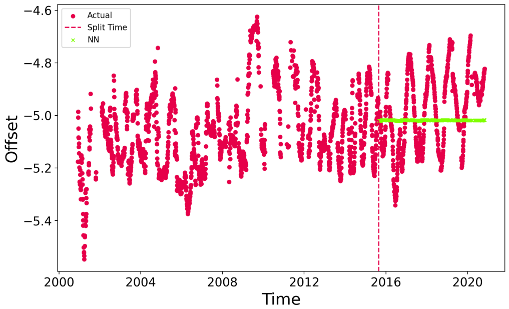

Fig. 11: The calibrated offsets of the Y axis of Cluster 3 (red) with the predicted offsets by an FNN model trained on 15 years of data (green). The data before the split time is the training data.

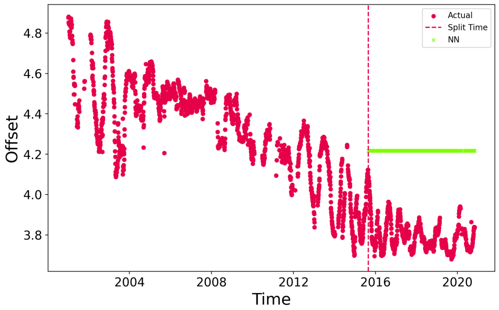

Fig. 12: The calibrated offsets of the Z axis of Cluster 4 (red) with the predicted offsets by an FNN model trained on 15 years of data (green). The data before the split time is the training data.

These results translate to a mean MAE of 0.21 ± 0.05, between the performance of the other two ML models. Notably, the FNN model for the Z axis of Cluster 4 appears to performs to an MAE of 0.43 ± 0.07 nT, significantly worse than the 0.09 ± 0.06 nT and 0.18 ± 0.11 nT from the KNN and SVR models for the same data. Examining the offset-time graphs reveals that some FNN models have not fully converged, causing this increased error.

The FNN predictions for Cluster 3's Y axis (Fig. 11) simply returned the training offsets' mean, indicating that the model has failed to learn useful patterns. This might imply the model is too simple to capture the patterns in the data. However, attempts to enhance model performance by adding layers, replacing the ReLU activation functions with leaky ReLU functions, varying the learning rate, and adding a 50% probability of dropout regularisation did not resolve the issue. This suggests that there is simply not enough training data for the neural network to identify trends.

Four of the eight spinning axes exhibited this underfitting problem. All of the underfit datasets are from Cluster 3 and 4. The reason why only Cluster 3 and 4, but not 1 and 2, were affected could not be ascertained.

This underfitting behaviour explains Cluster 4's Z axis' anomalously large MAE. As seen in Fig. 12, the model is underfit and will only return the mean of the training data. However, the offsets have a long term downwards trend, causing the testing offsets to be far from the mean of the training data.

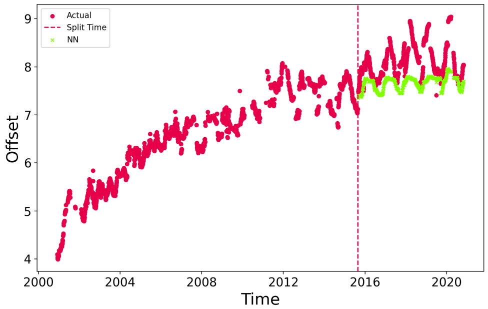

Fig. 13: The calibrated offsets of the Y axis of Cluster 1 (red) with the predicted offsets by an FNN model trained on 15 years of data (green). The data before the split time is the training data.

TABLE V: Mean Absolute Errors for LSTM Models

Cluster	Y	Z
1	6.16 ±0.35	0.67 ±0.24
2	1.86 ±0.91	2.45 ±0.52
3	5.62 ±0.78	2.40 ±0.67
4	2.47 ±0.71	5.88 ±0.26

Cluster 1's Y axis was one of the axes for which the FNN did not underfit (Fig. 13). The model behaves similarly to the SVR and KNN models: they predicted the general oscillations but struggled with the magnitudes of the peaks and troths and with predicting offsets out of the range seen in the training data.

To determine if the inability of the models to predict magnitudes correctly is due to time dependence in the data, it was decided to look at the orbits in the context of the previous orbits' offsets and features using an LSTM, a time-series-specific neural network.

VII. Long Short Term Memory

An LSTM neural network is a type of recurrent neural network useful for time series data because it retains information across several time steps, allowing it to learn time based dependencies. To be able to train the model on data from the previous orbits, 2D feature vectors were created that included the features of the previous nine orbits along with that of the orbit being predicted.

Using the Keras library for Python, a bidirectional LSTM, a type of LSTM that processes the time series data in both directions to better capture dependencies in the data, was implemented with a single LSTM layer. The number of neurons in the layer was optimised by training LSTM models with 32, 64, 128, and 256 neurons for each rotating axis and selecting the one with the lowest MAE. These MAEs are listed in Table V.

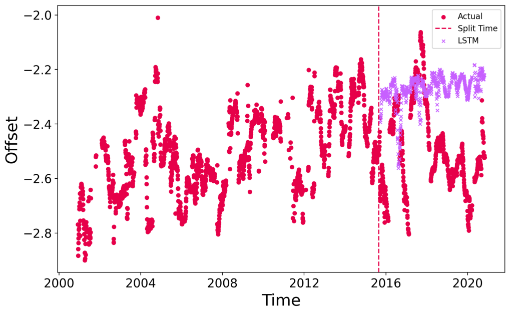

Fig. 14: The calibrated offsets of the Y axis of Cluster 3 (red) with the predicted offsets by an LSTM model trained on 15 years of data (purple). The data before the split time is the training data.

TABLE VI: Mean Mean Absolute Errors by Calibration Technique

	Error (nT)
Manual	0.05
KNN	0.19 ± 0.05
SVR	0.24 ± 0.07
FNN	0.21 ± 0.05
LSTM	3.44 ± 0.26

Overall, the LSTM models achieved a mean MAE of 3.44 ± 0.26, significantly worse than the previous models due to underfitting similar to the FNNs. Attempts to improve performance by incorporating previous offset values and the rate of change of offset of the other rotating axis on the spacecraft did not resolve the underfitting. This indicates that, while the features may carry relevant information, there are insufficient training datapoints for the LSTM to learn meaningful patterns in the complex, high-dimensional dataset.

When the LSTM models trained on sequences of the standard two temperature feature vectors and converged, they behaved similarly to the previous models: they predicted the oscillations but struggled with the magnitudes of the offsets. Even for noisy data, such as the large offset jumps in Cluster 3's Y axis (Fig. 14), the model can still capture upwards or downwards trends, yet the peak and trough values deviated significantly from the actual values.

VIII. Discussion

The primary objectives of this investigation were to use spacecraft telemetry to predictively calibrate the magnetometer offsets and to improve the interpolation of the X axis offsets between calibrations. Based on the performance of the models trained and tested during this project, several conclusions can now be drawn.

The ML models underperformed compared to manual calibration, exhibiting errors of at least four times greater than the 0.05 nT error of present methods. Table VI summarises each model's mean MAE.

The performance of the models in this project demonstrates that model MAE scales with the range of variation in the training data. This is evident from the noticeably larger MAE in the predicted offsets for Cluster 1 compared to the other spacecraft for the same models, driven by the larger variations in Cluster 1's offsets. Consequently, for optimal results in ML calibration tasks, it is crucial to minimise the absolute variation of the calibrated value.

The calibrated reference offsets displayed annual oscillations. While the models generally predicted the trends of these oscillations accurately, they struggled to predict the amplitudes. This difficulty was particularly apparent during periods of rapid offset changes, such as around 2020 in Fig. 7. The models' failure to predict the magnitudes implies that the offsets cannot be fully characterised by the two temperature features alone; there must be some other variable affecting the offsets.

All the variables from the available telemetry that were expected to affect the offset were already examined. However, it remains possible that the disregarded telemetry values could affect the offset in ways which were not immediately apparent. Theoretically, feeding all available data into the FNN would allow the model to identify the meaningful features and discard the rest. However, the FNN models failed to converge with even two dimensional feature vectors, indicating that there are insufficient data points for the FNN model to learn the patterns in the data. As such, using an FNN to identify additional meaningful variables would be infeasible without more data. Unfortunately, obtaining additional data would involve calibrating the spin axis offsets on a sub-orbital basis, which is not practical.

Generally, the models also struggled to predict offsets outside the range of the training data. This limitation is inherent to ML-based approaches, which can rely solely on patterns in the training data. Addressing this issue would require that instruments be extensively tested on the ground to find which variables affect offsets and how before they are launched.

The ability of these models to predict the trend of the annual oscillations has potential for improving the interpolation of the X axis between the calibrations. However, because of limited data for the X axis offsets and that the models already struggle to converge for the Y and Z axes, it is unlikely that a model could be trained to interpolate the values.

The offset jumps observed after restarts pose another problem as none of the models could follow them, which implies that they are either random or affected by another variable. Until the causes for these offset jumps can be ascertained, ML models will not be able to match the performance of manual calibration techniques.

IX. Conclusions and Recommendations

Overall, the models predicted the offsets with errors at least four times greater than the error yielded by regular calibration. The magnitude of prediction error yielded by the models was proportional to range of the training data. Key limitations included the inability to predict offsets outside the training range, failure to capture oscillation amplitudes and reset jumps using the two temperature features, and convergence issues in some models.

Unfortunately, mitigating these sources of error would require non-trivial effort. Because the absolute prediction errors scaled with the range of the training data, an ideal instrument for this kind of ML calibration would have to have very stable offsets. These stable offsets are also a requirement such that the range of offsets to be predicted will always have been seen in training. The additional variables that affect the amplitude of the offsets and jumps would also have to be identified. An easier way of accomplishing this would be to feed many features into a neural network and let it discover what features are meaningful. However, this is only possible with more data. The models were trained using thousands training points; even then, the neural network would often not converge with two features. These thousands of data points were collected over two decades: there are not many missions where there is more data available than for Cluster. Given this limitation in the amount of data, it can be concluded that these ML techniques are not well suited to spacecraft instrument calibration.

REFERENCES

[1] *Accurate determination of magnetic field gradients from four point vector measurements. I. Use of natural constraints on vector data obtained from a single spinning spacecraft*. ieeexplore.ieee.org. URL: https://ieeexplore.ieee.org/document/486522/authors#authors (visited on 10/10/2023).