A Real-Time Anomaly Detection in Satellite Telemetry Data Using Artificial Intelligence Techniques Depending on Time-Series Analysis

Dr. Mohamed Ahmed Hussein Ali
Computer Science Department
High Institute for Computers and Information Technology
AL-Shorouk Academy
Cairo – Egypt
email: dr.mohamed.hussein@sha.edu.eg

Abstract: To effectively detect and identify the anomaly data in massive satellite telemetry data sets, the novel detection and identification method based on the Auto-regressive integrated moving average (ARIMA), Prophet, Long Short Term Memory (LSTM), and Auto-encoder algorithms were proposed in this paper. The proposed model is used to find anomalous events by comparing the actual observed values with the predicted intervals of telemetry data.

First, preprocessing for the raw telemetry data were Handled for the missing values using linear interpolation. Second, Down-casting to reduce the memory storage. Based on this symbolization, the pseudo-period of the data was extracted. Third, the Data Transformation and Scaling to normalize the data within a particular range to helps in speeding up the calculations were applied. Finally, the experimental results for the Prophet model show predictions with high efficiency, stable when detecting anomalies, and requires little computational time. The results of Prophet compared with other applied algorithms, demonstrate the effectiveness and superiority of the proposed model.

I. INTRODUCTION

Satellite consists of many subsystems like Power subsystem, Command and data handling subsystem CDHS, Communication subsystem, Thermal control subsystem, attitude determination and control subsystem, Telemetry, tracking, and command (TT&C), Propulsion subsystem and payload subsystem.

Current anomaly detection methods for spacecraft telemetry primarily consist of tiered alarms indicating when values stray out- side of pre-defined limits and manual analysis of visualizations and aggregate channel statistics. Expert systems and nearest neighbor- based approaches have also been implemented for a small number of spacecraft [13]. These approaches have well-documented limitations – extensive expert knowledge and human capital are needed to define and update nominal ranges and perform ongoing analysis of telemetry. Statistical and limit-based or density-based approaches are also prone to missing anomalies that occur within defined limits or those characterized by a temporal element [9].

The massive amounts of telemetry data transmitted by an in-orbit satellite are the sole observational basis of the satellite's operation. Through the analysis of telemetry data, ground telemetry, track, and command stations can determine the satellite's operational state and detect possible anomalies in a timely fashion, assisting the normal operation of the in-orbit satellite.

Challenges central to anomaly detection in multivariate time series data also hold for spacecraft telemetry. A lack of labeled anomalies necessitates the use of unsupervised or semi-supervised approaches. Real-world systems are usually highly non-stationary and dependent on current context. Data being monitored are often heterogeneous, noisy, and high-dimensional. In scenarios where anomaly detection is being used as a diagnostic tool, a degree of interpretability is required.

The breadth and depth of research in anomaly detection offers numerous definitions of anomaly types, but with regard to time- series data it is useful to consider three categories of anomalies – ***point, contextual***, and ***collective*** [9]. Point anomalies are single values that fall within low-density regions of values, collective anomalies indicate that a sequence of values is anomalous rather than any single value by itself, and contextual anomalies are single values that do not fall within low-density regions yet are anomalous with regard to local values.

Simple forms of anomaly detection consist of out-of-limits (OOL) approaches which use predefined thresholds and raw data values to detect anomalies. However, due to the influence of complex noise in the actual telemetry data, the fixed threshold method is prone to producing false alarms in the detection. In addition, the method cannot detect anomalies within the threshold. A myriad of other anomaly detection techniques has been introduced and explored as potential improvements over OOL approaches, such as clustering-based approaches [15, 24, 28], nearest neighbors approaches [3, 6, 23, 25], expert systems [7, 34, 36, 43], and dimensionality reduction approaches [14, 39, 45], among others. These approaches represent a general improvement over OOL approaches and have been shown to be effective in a variety of use cases, yet each has its own disadvantages related to parameter specification, interpretability, generalizability, or computational expense [9, 16] (see [9] for a survey of anomaly detection approaches). Recently, RNNs have demonstrated state-of-the-art performance on a variety of sequence-to-sequence learning benchmarks and have shown effectiveness across a variety of domains [38]. In the following sections, we discuss the shortcomings of prior approaches in aerospace applications and demonstrate RNN's capacity to help address these challenges.

Historical telemetry data have been used for modeling, with the measured data compared with the predicted data of the proposed model in order to achieve the anomaly detection and identification of the measured data [3–5]. The performance of this method is directly related to the modeling accuracy. When the data type changes, the data model must be updated. Based on the threshold method, Song [6] used the semi-major axis change method (SACM) to extract the mean and standard deviation of telemetry data in different periods as the migration variables for anomaly detection and identification. However, this method showed limited accuracy for anomaly identification. To diagnose the satellite faults, Sherr [7] used empirical mode decomposition to extract the telemetry data features from the time-frequency domain. However, the characteristic frequencies of each component in the data must be determined first, which is difficult in practical application.

Due to their limited feasibility in use for the diagnosis of anomalous satellite telemetry data, the methods described above cannot meet engineering requirements.

This paper proposes a development of a software for telemetry-data analysis and forecasting to provide progressive solutions for better satellite management which improves the use of satellite resources and increases satellite lifetime.

II. Time Series Models

A time series is a sequence where a metric is recorded over regular time intervals. Depending on the frequency, a time series can be of yearly, quarterly, monthly, weekly, daily, hourly, minutes and even seconds wise.

A. ARIMA

ARIMA stands for "Auto-Regressive Integrated Moving Average". ARIMA models provide an approach to time series prediction and provide complementary approaches to the problem. It refers to a set of models that explain a time series based on its past values, that is, its lag and lagging prediction errors so that this equation can be used to predict future values. It aims to describe automatic correlations in the data. ARIMA models can be used to model any non-seasonal time series that have patterns and are not random white noise. It's characterized by three terms: p, d, q. Before introducing ARIMA models in detail, it must first discuss the concept of Stationarity and the technique of time-series Differencing.

A stationary time series is a series whose statistical properties do not depend on the time at which the series is observed, that is, its mean and variance. Hence, trends or seasonal time series are not stationary. The trend and seasonality will affect the time series value at different times. The stationary is very important because it is easy to make predictions on a stationary series as we can assume that future statistical properties will not differ from those currently observed. Future projections will be wrong if the original series were not stationary.

Differencing is a statistical approach to making a time series dataset stationary by transforming it. It can be used to remove the series dependence on time, so-called temporal dependence. Trends and seasonality are examples of such structures. By removing variations in the level of a time series, differencing can assist in stabilizing the mean of the time series, hence eliminating (or reducing) trend and seasonality.

B. Prophet

Prophet is a procedure for forecasting time series data based on an additive model where non-linear trends are fit with yearly, weekly, and daily seasonality, plus holiday effects. It's a curve-fitting algorithm to build time series, forecasting models. It is open-source software released by Facebook's Core Data Science team. It designed for ease of use without expert knowledge on time series forecasting or statistics. It Builds a model by finding the best smooth line which can be represented as the sum of three component; Overall growth trend, Yearly seasonality and Weekly seasonality.

Prophet library was launched by Facebook as an API for carrying out the forecasting related things for time series data. The library is so powerful that it has the capability of handling stationarity within the data and seasonality-related components.

C. Long Short-Term Memory

Long Short-Term Memory networks – usually just called "LSTMs" – are a special kind of RNN, capable of learning long-term dependencies. They were introduced by Hoch Reiter & Schmid Huber (1997) and were refined and popularized by many people in the following work. The Long Short-Term Memory network, or LSTM network, is a recurrent neural network that is trained using Backpropagation Through Time and overcomes the vanishing gradient problem. As such, it can be used to create large recurrent networks that in turn can be used to address difficult sequence problems in machine learning and achieve state-of-the-art results.
Instead of neurons, LSTM networks have memory blocks that are connected through layers. A block has components that make it smarter than a classical neuron and a memory for recent sequences. A block contains gates that manage the block's state and output.

D. Auto Encoder

Auto-encoders are neural networks. the defining aspect of an auto-encoder is that the input layers contain exactly as much information and have the exact same number of units as the output layer. The reason is that an auto-encoder aims to replicate the input data. It outputs a copy of the data after analyzing it and reconstructing it in an unsupervised technique. Auto-encoders operate by taking in data, compressing, and encoding the data, and then reconstructing the data from the encoding representation. The model is trained until the loss is minimized and the data is reproduced as closely as possible. Through this process, an auto-encoder can learn the important features of the data. The data that moves through an auto-encoder is not just mapped straight from input to output, meaning that the network does not just copy the input data.

III. PROPOSED SYSTEM

This paper proposes an automated anomaly detection for satellite telemetry data using Artificial Intelligence techniques that depend on time-series analysis to detect possible future problems for handling them. The proposed system will predict the anomaly in satellite subsystems by monitoring and analysis the satellite telemetry data to protect the satellite system from any

harm that could happen. The proposed system detects the pattern of satellite telemetry stream then simulate the satellite data streaming process.
Received telemetry data contain two types of data must get rid of them which are outlier data (Extreme values) and anomaly (referred to the identification of events that do not conform to an expected pattern).
However, if the data collected includes, for example, tensor (multi-way) structure, space-time measurement values, such as the measurement of satellite subsystems, some significant anomalies may stay hidden. Considering that the proposed system aims to manage satellite telemetry data, so an anomaly/outlier detection system is necessary to monitor and characterize the telemetry data.
The system is being developed to capture data from satellites, considering the cadence of the data (the regular frequency at which measurements are taken) is one measurement every T minutes. The developed system aimed to forecast time series data and detect anomalies.

A. *System Design*

The design of proposed system went through, software architecture design and user interface design.

The Satellite Anomaly Detection system receives the telemetry data from the satellite in the LEO and then provides feedback/status to the FCC based on the input, in this case, to give a warning that something may go wrong. Using an adapted time series model that detect normal pattern then classify anomalies based on prediction error. Also, the Satellite Anomaly Detection system can forecast future telemetry values for a specified period in the future.

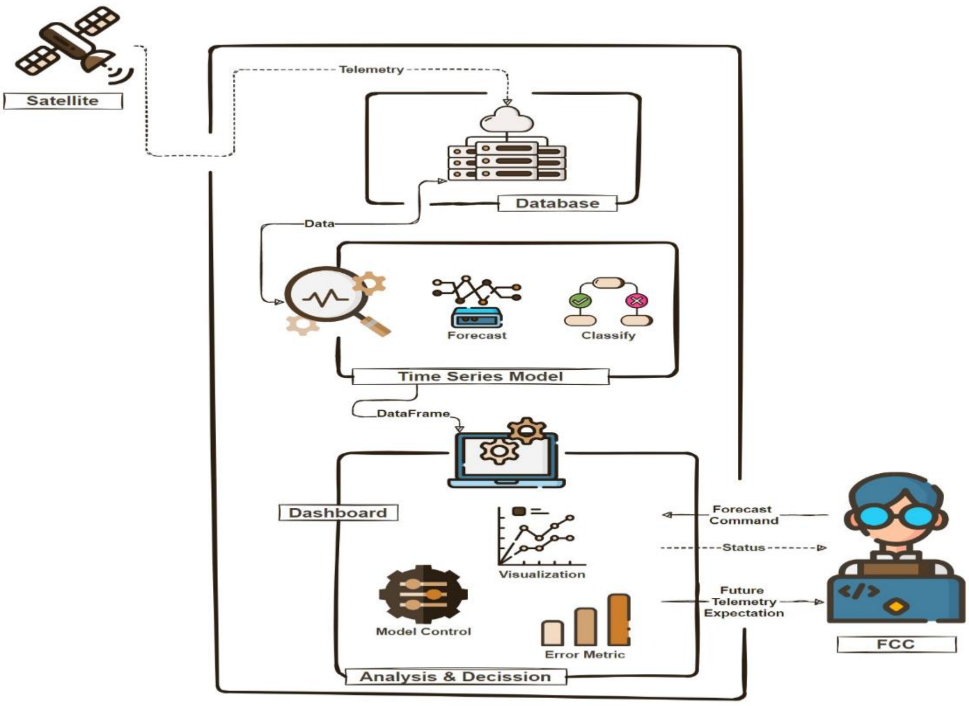

Figure 1.System architecture

The designed user interface included control card of the model to represent the capture interval (daily, weekly, or yearly seasonal) and represent the selected interval: week, month, or year to updates the model and data.

It is also designed data streaming card to plot telemetry data points in real-time and to give a clear visualization of data patterns, and anomalies with their impacts, also to visualize model forecasts. Finally, design dashboard interface to summarize all operations of the proposed system.

B. implementation

The implementation process is to study a satellite data in low earth orbit then developing a real-time dashboard that can detect anomalies in real time. The implementation, including code, system architecture as well as graphs and results.

- **Dataset** comes from NASA's satellite in low earth orbit that is using sensors for studying purposes. This satellite lifetime around 20 years, data have been collected over this period, which is just a value of subsystem measurements called Telemetry Data. This research chooses the most critical subsystem datasets the influence directly in the satellite life time. The chosen datasets are battery temperature, total spacecraft bus, bus voltage and reaction wheel RPM.

  Battery Temperature Dataset: It is a **univariate** time series with 1,518,180 readings for battery temperature recorded over roughly 15 years. "Univariate" means that it tracks only one variable's values over time. The cadence of the data is one measurement every 5 minutes.

  Total Spacecraft Bus Current Dataset: A satellite bus or spacecraft bus is a general model on which multiple-production satellite spacecraft are often based. The bus is the infrastructure of the spacecraft, usually providing locations for the payload.
  Bus Voltage Dataset: It is a univariate time series with 1,838,088 readings for bus voltage recorded over roughly 18 years. The cadence of the data is one measurement every 5 minutes.
  Reaction Wheel RPM Dataset: It is a univariate time series with 48,865,494 readings for RPM recorded over roughly 10 years. The cadence of the data is one measurement every second. A reaction wheel is sometimes operated as (and referred to as) a momentum wheel, by operating it at a constant (or near-constant) rotation speed, to imbue a satellite with a large amount of stored angular momentum.

- **Data Preprocessing**

  The dataset sometimes not complete due to not all subsystems are working all the time (like reaction wheel), or weakness of signal or loss of communications.
  Linear interpolation is the simplest method of getting values at positions in between the data points. The points are simply joined by straight line segments. Know the formula for the linear interpolation process. The formula is

$$y(x) = y1 + (x - x1)\frac{(y2 - y1)}{(x2 - x1)}$$

  where x is the known value, y is the unknown value, x1 and y1 are the coordinates that are below the known x value, and x2 and y2 are the coordinates that are above the x value. In the mathematical field of numerical analysis, interpolation is a method of constructing new data points within the range of a discrete set of known data points. A few data points from the original function can be interpolated to produce a simpler function that is still fairly close to the original.

  Due to the high sampling rates of physical sensors, a time series reduction is applied. Resampling and Peak Detection approaches have been applied to solve this problem while the key characteristics of the data remain unchanged.

- **Modeling**

This phase was for implementing and applying the different time series models: ARIMA, prophet, LSTM and Autoencoder models on the data set for different satellite subsystems.

Arima model on Reaction Wheel Temperature Data Set

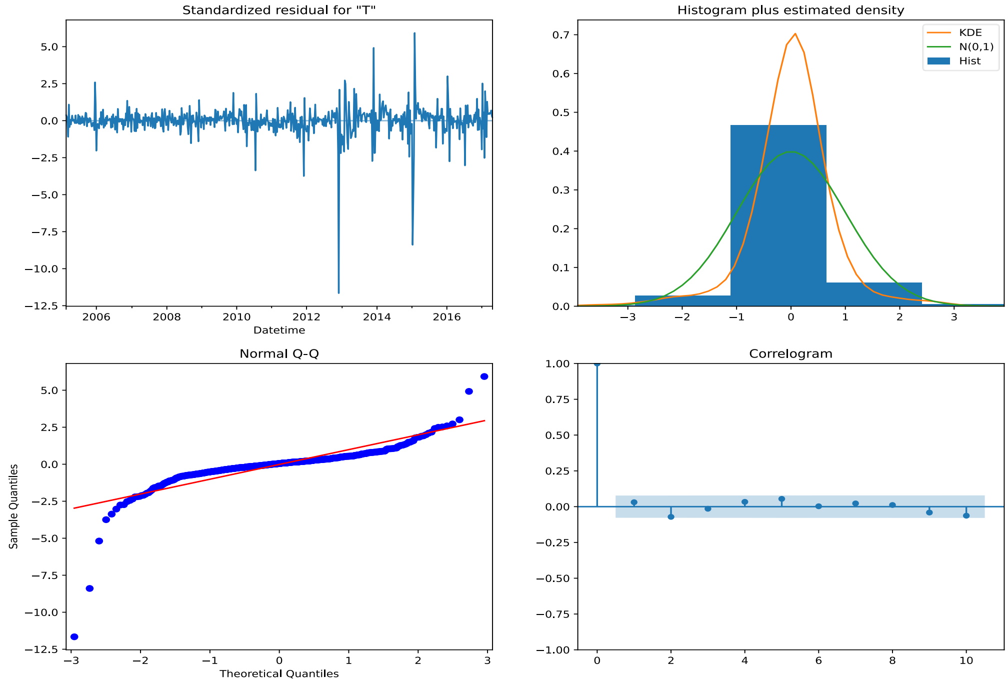

Figure 2 ARIMA Model on Data stream

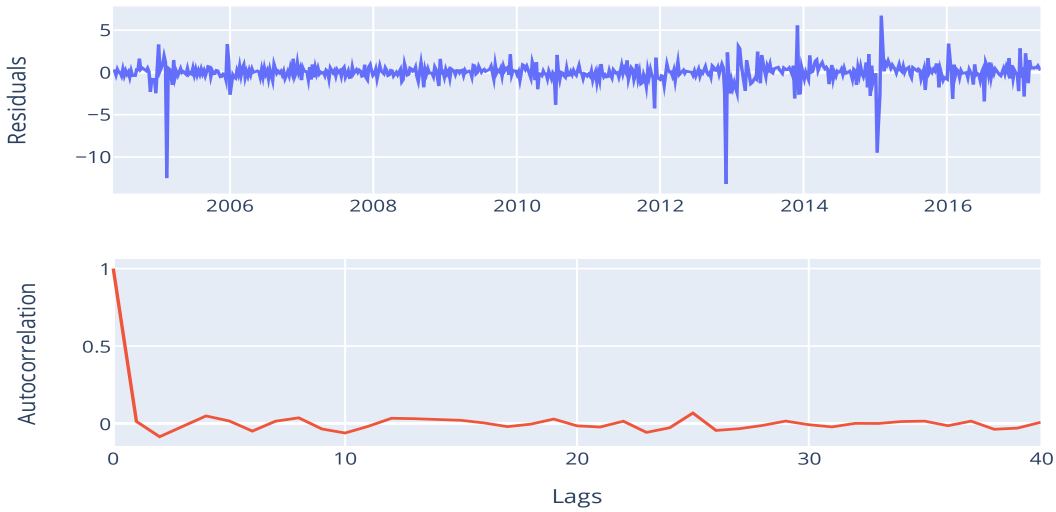

Figure 3 Residuals

- **Dashboard implementation**

The implemented dashboard is to perform a comparison between the real streaming from the satellite and the database.

C. Testing

- **Stationarity Test**

Stationarity Test by using **Plotting Rolling Statistics** to find rolling mean and variance to check stationary. Another way to check stationarity by using **Dickey-Fuller Test.**

-Results for **Dickey-Fuller Test**

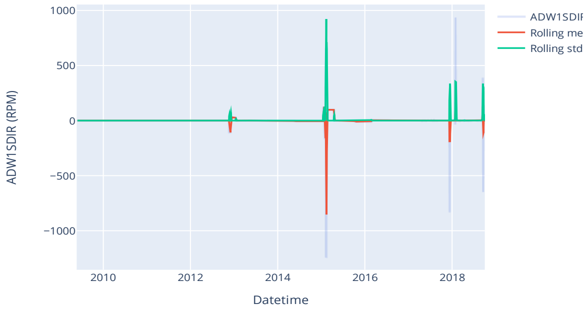

Figure 4 Rolling statistics plot for reaction wheel RPM

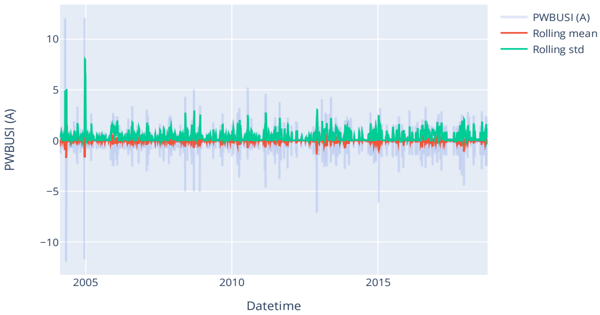

Figure 5 Rolling statistics for total spacecraft bus current

Figure 6 Rolling statistics for battery Temp

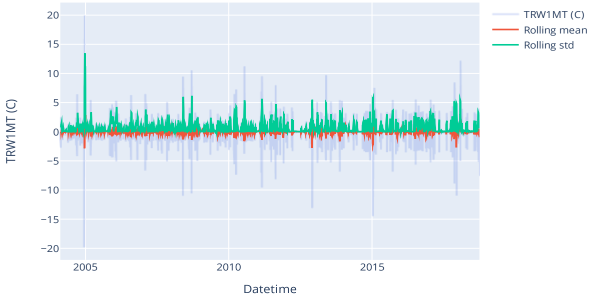

Figure 7 Rolling statistics for wheel temp

Figure 8 Rolling statistics for bus voltage

- ***Train test split***

Splitting dataset into training and testing subsets. The model can be prepared on the training dataset and predictions can be made and evaluated for the test dataset. This can be done by selecting an arbitrary split point in the ordered list of observations and creating two new datasets.

- ***Multi Train Test Split***

A repetition of the process of splitting the time series into train and test sets multiple times will require multiple models to be trained and evaluated, but this additional computational expense will provide a more robust estimate of the expected performance of the chosen method and configuration on unseen data. This can done manually by repeating the process described in the previous section with different split points.

D. EXPERIMENTAL WORK

For many spacecrafts, current anomaly detection systems are difficult to assess. The precision and recall of alarms aren't captured and telemetry assessments are often performed manually. At this section we do some experiments using the proposed model on data like (Reaction Wheel Temperature, Battery Temperature, Reaction Wheel Temperature, Bus Voltage and Reaction Wheel RPM), and Observed Anomalies/Outliers .

Experiment 1 Reaction Wheel Temperature

Table 1. model parameters and results reaction wheel temperature

Training Size	Test Size	Valid Size	Cadence	Model	Time Consumed	Scaler	Optimizer	Loss	Epochs
689	76	0	Week	ARIMA	56min	None	AIC	MSE	None
4811	535	0	Day	Prophet	8.45 s	None	Stan	MSE	None
3849	535	962	Day	LSTM	30min 41s	MinMaxScaler	Adam	MSE	25
3367	535	1443	Day	Autoencoder	3min	MinMaxScaler	Adam	MSE	25

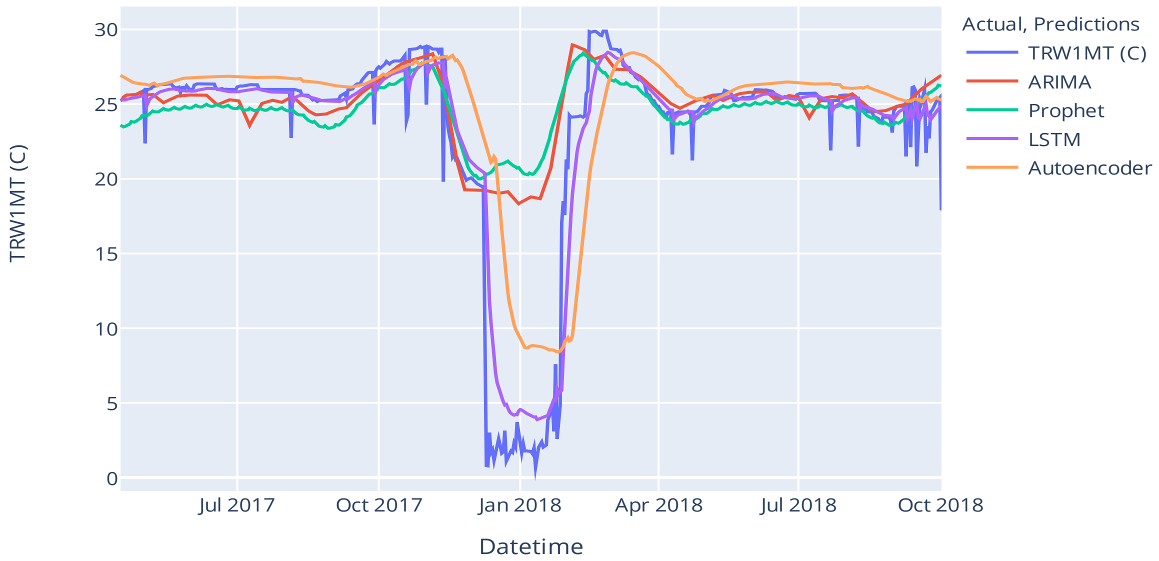

Figure 91. Reaction Wheel Temperature Forecast

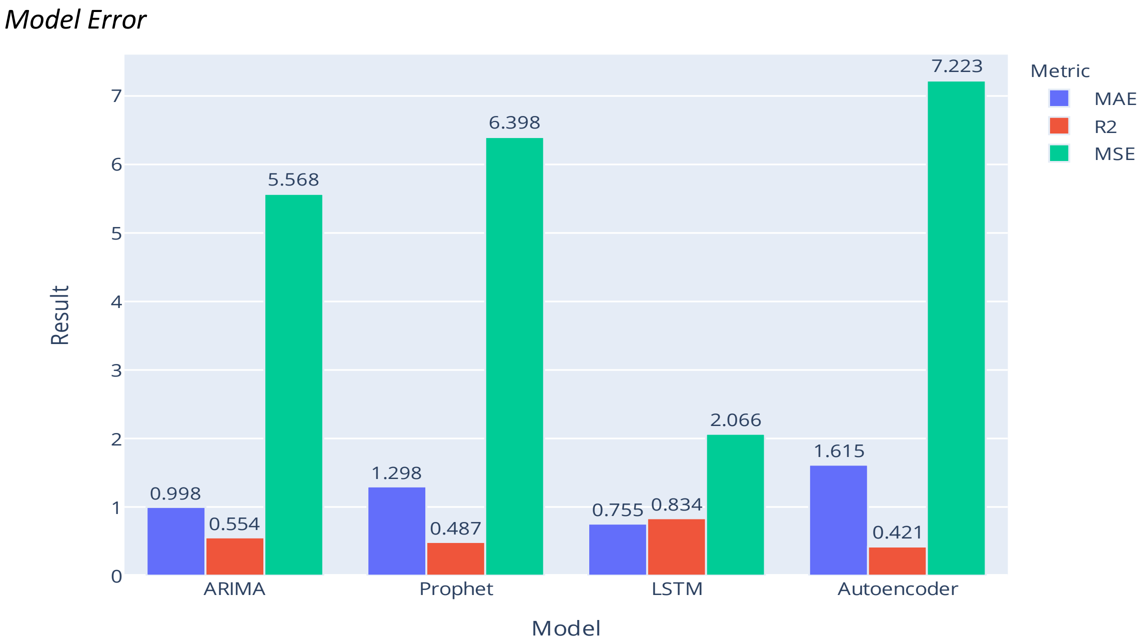

Figure 2.Reaction Wheel Temperature Forecast Error

<u>Experiment 2</u> Battery Temperature

Table 2. model parameters and results battery temperature

Training Size	Test Size	Valid Size	Cadence	Model	Time Consumed	Scaler	Optimizer	Loss	Epochs
689	76	0	Week	ARIMA	1h 9min 50s	None	AIC	MSE	None
4811	535	0	Day	Prophet	8.9 s	None	Stan	MSE	None
3849	535	962	Day	LSTM	34min 25s	MinMaxScaler	Adam	MSE	25
3340	535	1431	Day	Autoencoder	2min 6s	MinMaxScaler	Adam	MSE	25

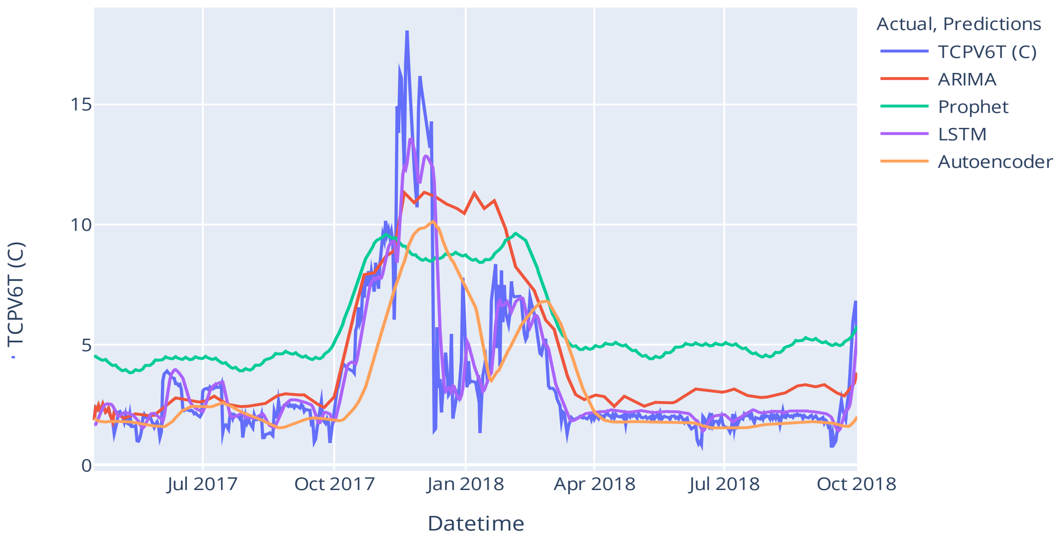

Figure 11.Battery Temperature Forecast

__Model Error__

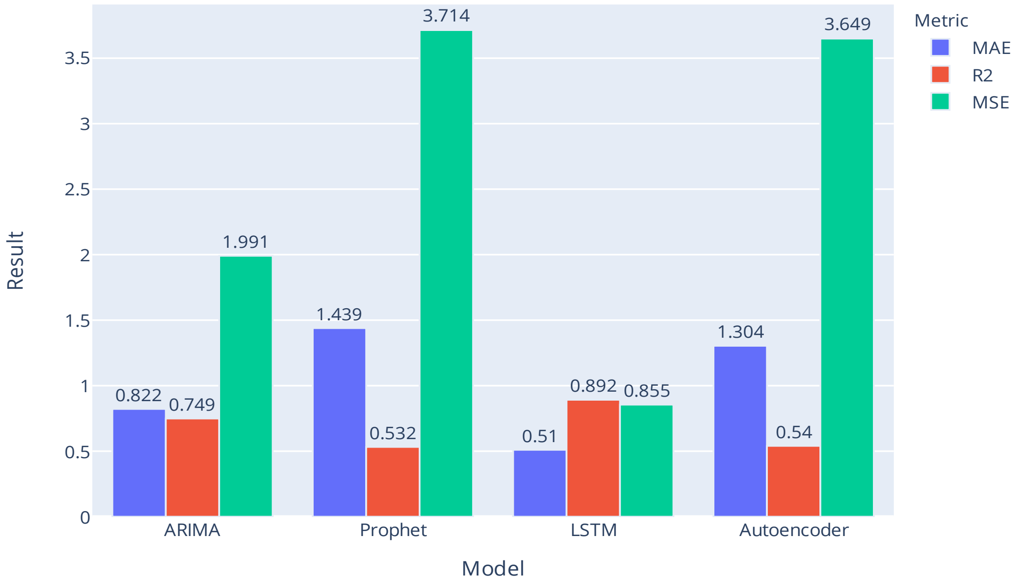

Figure 12.Battery Temperature Forecast

<u>Experiment 3</u> Reaction Wheel RPM

Training Size	Test Size	Valid Size	Cadence	Model	Time Consumed	Scaler	Optimizer	Loss	Epochs
441	49	0	Week	ARIMA	55min 23s	None	AIC	MSE	None
3079	342	0	Day	Prophet	8.85 s	None	Stan	MSE	None
2463	342	616	Day	LSTM	22.6 s	MinMaxScaler	Adam	MSE	25
2128	342	911	Day	AE	1min 11s	MinMaxScaler	Adam	MSE	25

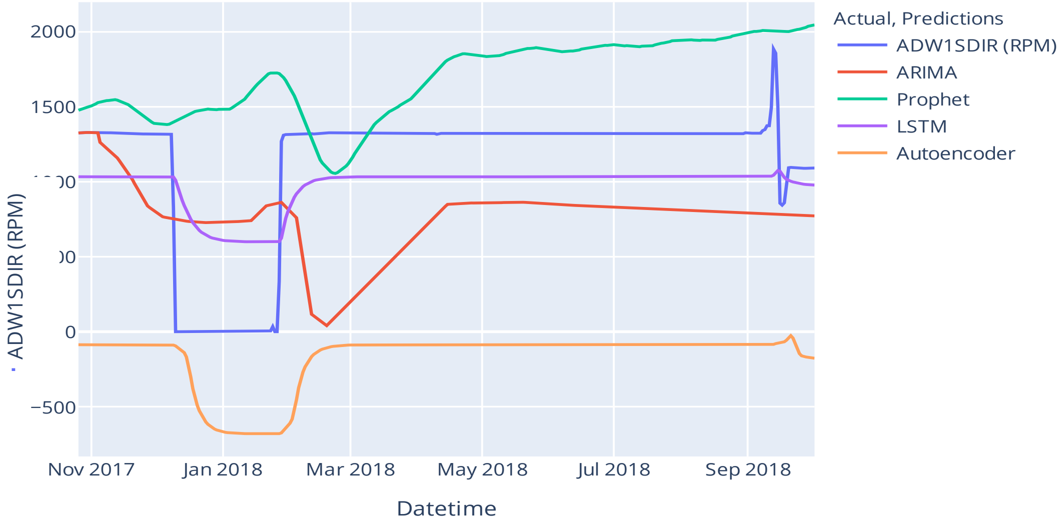

Figure 13. Reaction wheel RPM Forecase

Model Error

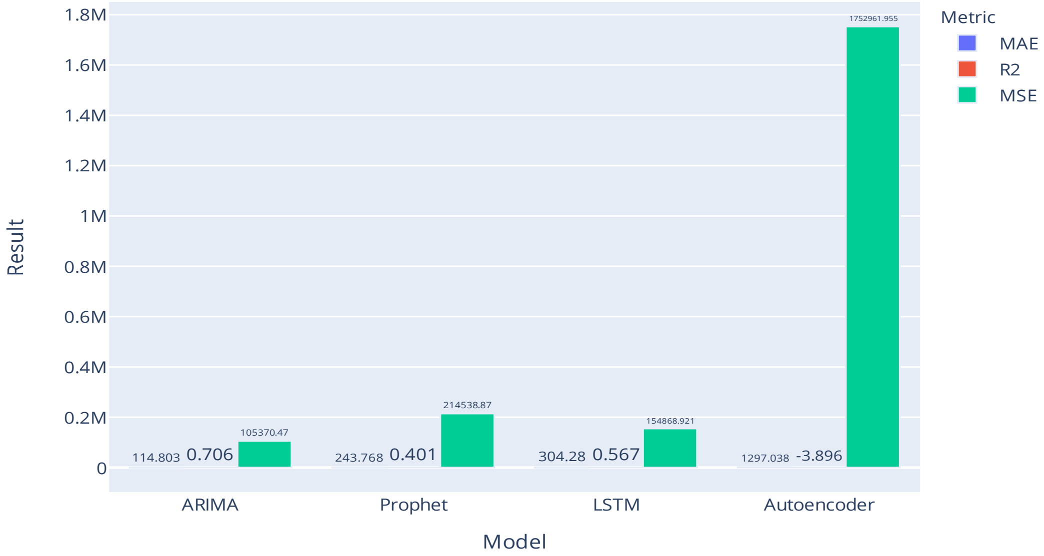

Figure 3. Reaction wheel RPM Forecase Error

Experiment 4 Total Spacecraft Bus Current

Training Size	Test Size	Valid Size	Cadence	Model	Time Consumed	Scaler	Optimizer	Loss	Epochs
689	76	0	Week	ARIMA	1h 8min 11s	None	AIC	MSE	None
4811	535	0	Day	Prophet	12.3 s	None	Stan	MSE	None
3849	535	962	Day	LSTM	3min 6s	MinMaxScaler	Adam	MSE	25
3367	535	1443	Day	AE	2min 2s	MinMaxScaler	Adam	MSE	7

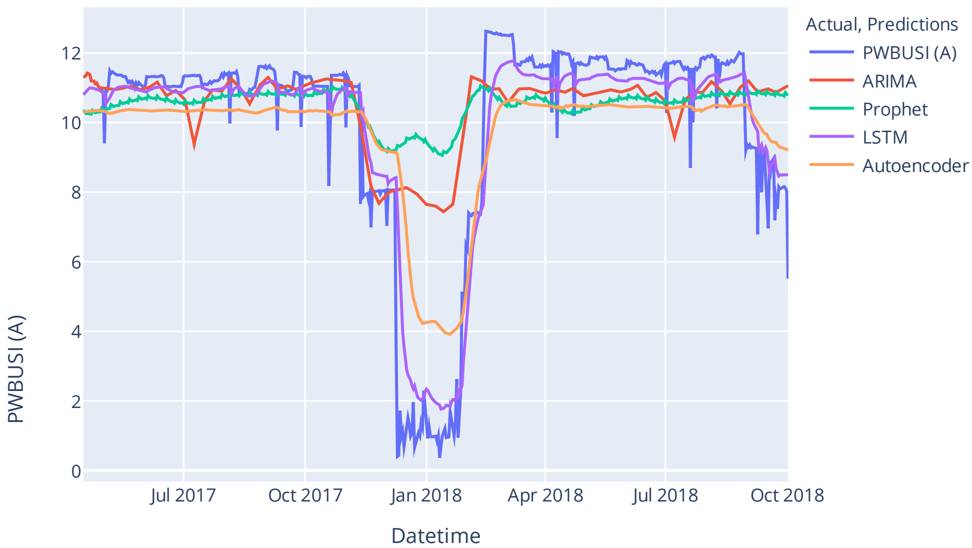

Figure15.Total Spacecraft Bus Current Forecast

Model Error

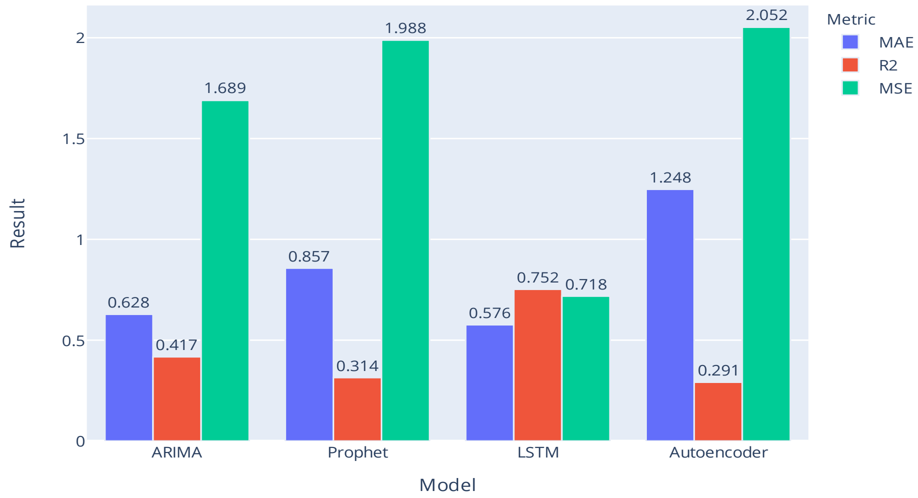

Figure 16.Total Spacecraft Bus Current Forecast Error

<u>Experiment 5</u> Bus Voltage

Training Size	Test Size	Valid Size	Cadence	Model	Time Consumed	Scaler	Optimizer	Loss	Epochs
689	76	0	Week	ARIMA	1h 30min 36s	None	AIC	MSE	None
5818	646	0	Day	Prophet	13.4 s	None	Stan	MSE	None
4654	1164	646	Day	LSTM	43.4 s	MinMaxScaler	Adam	MSE	5
4045	646	1733	Day	AE	5min 34s	MinMaxScaler	Adam	MSE	20

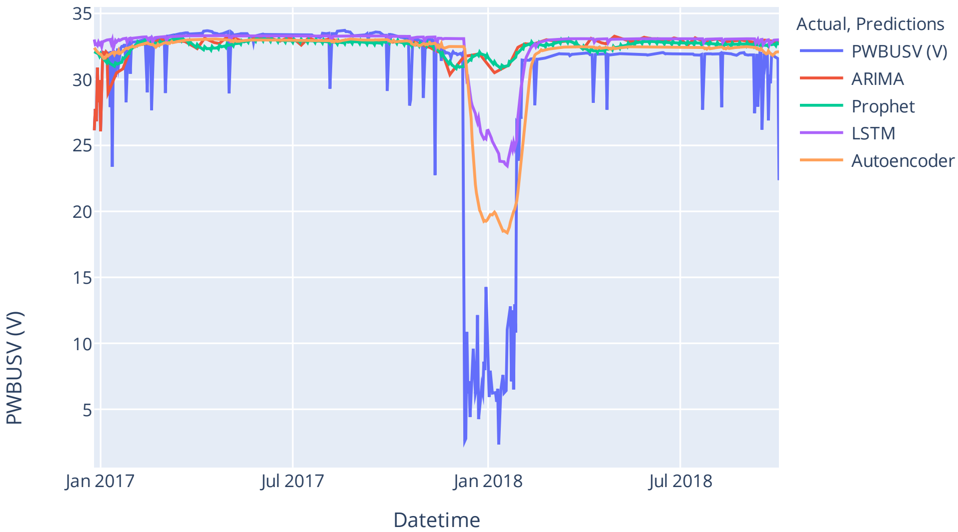

Figure 17.Bus Voltage Forecast

<u>Model Error</u>

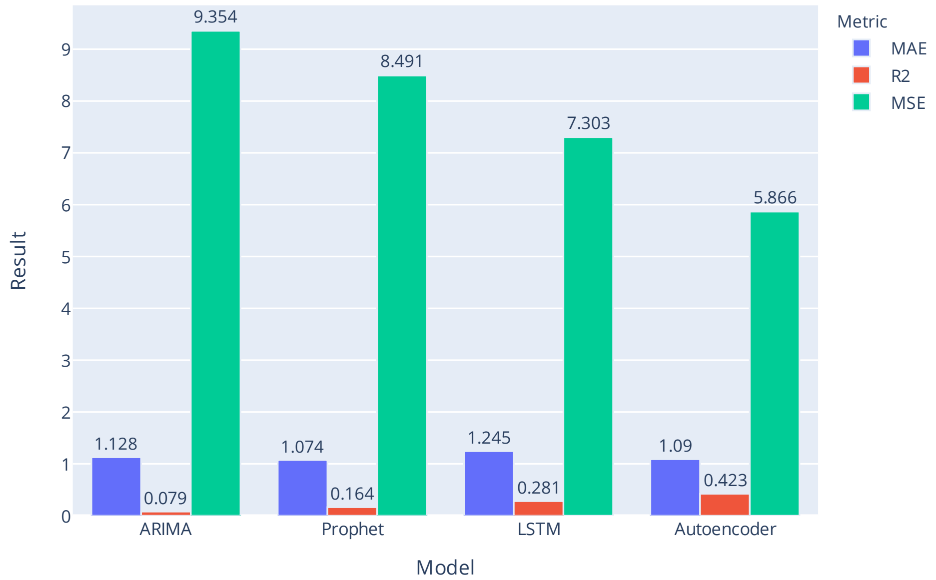

Figure 18.Bus Voltage Forecast

Observed Anomalies/Outliers

Classified Results

	Total Observations	Normal Observations	Anomaly Impact 1	Anomaly Impact 2	Anomaly Impact 3
TCPV6T (C)	5346	5056	247	43	0
ADW1SDIR (RPM)	3421	3331	55	20	15
TRW1MT (C)	5346	5243	29	67	7
PWBUSV (V)	6464	6322	53	52	37
PWBUSI (A)	5346	5214	49	76	7

Reaction Wheel Temperature

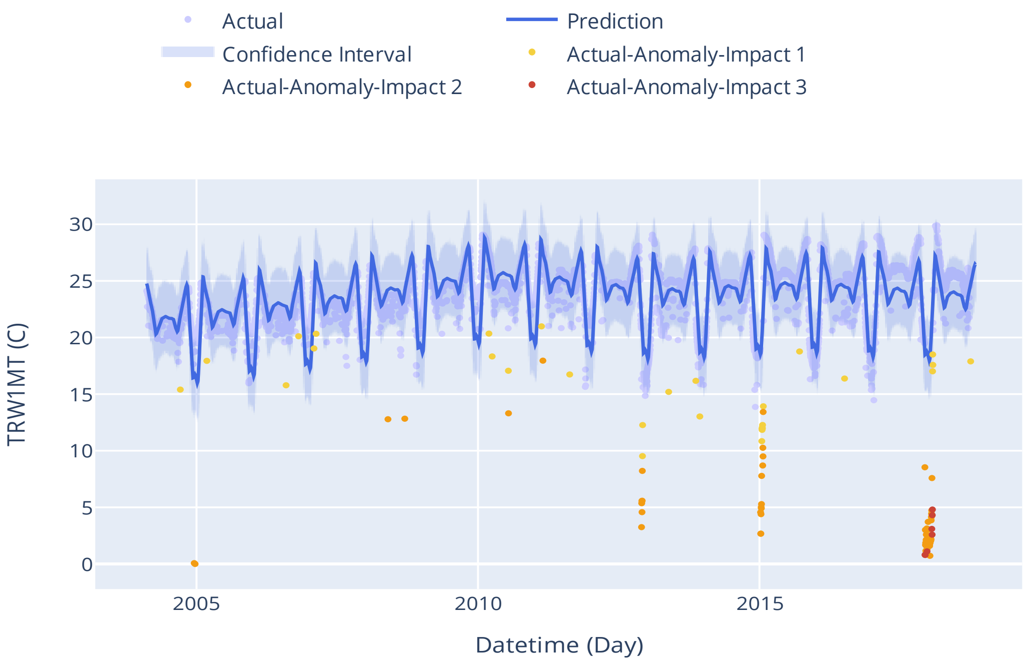

Figure 19. Observed Anomalies/Outliers Reaction Wheel Temperature

Battery Temperature

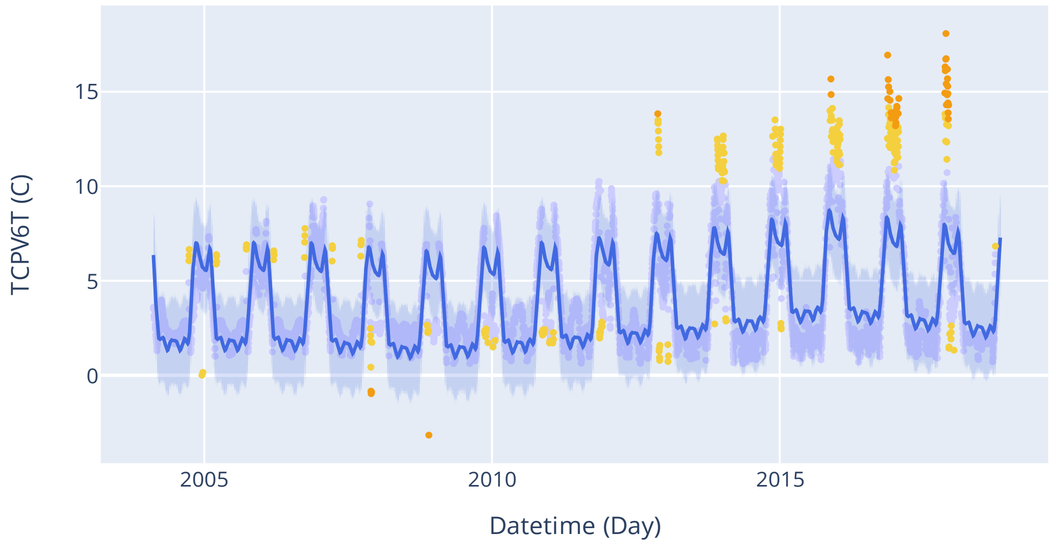

Figure 20. Observed Anomalies/Outliers Battery Temperature

Reaction Wheel RPM

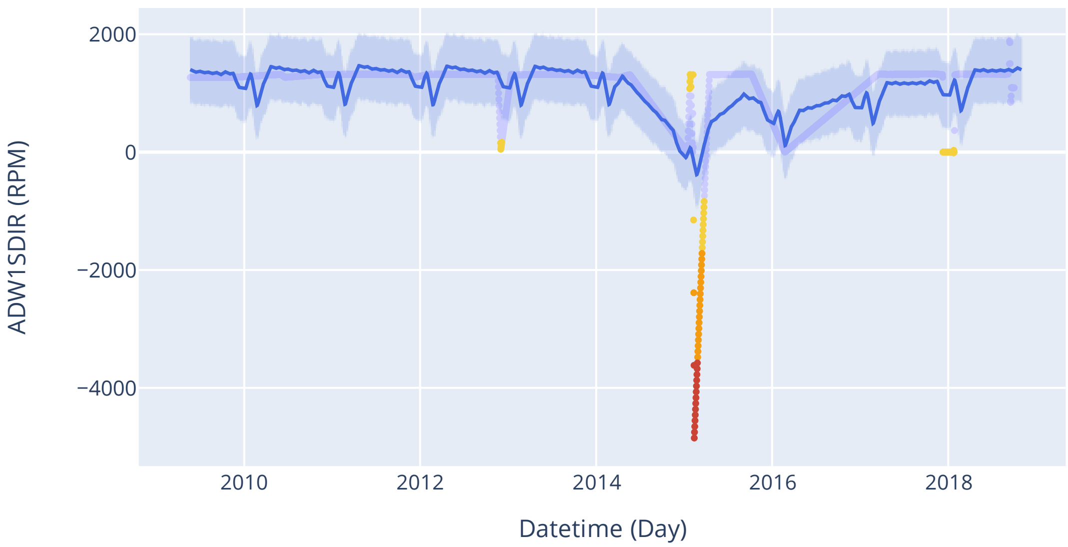

Figure 4Observed Anomalies/Outliers Reaction Wheel RPM

Total Spacecraft Bus Current

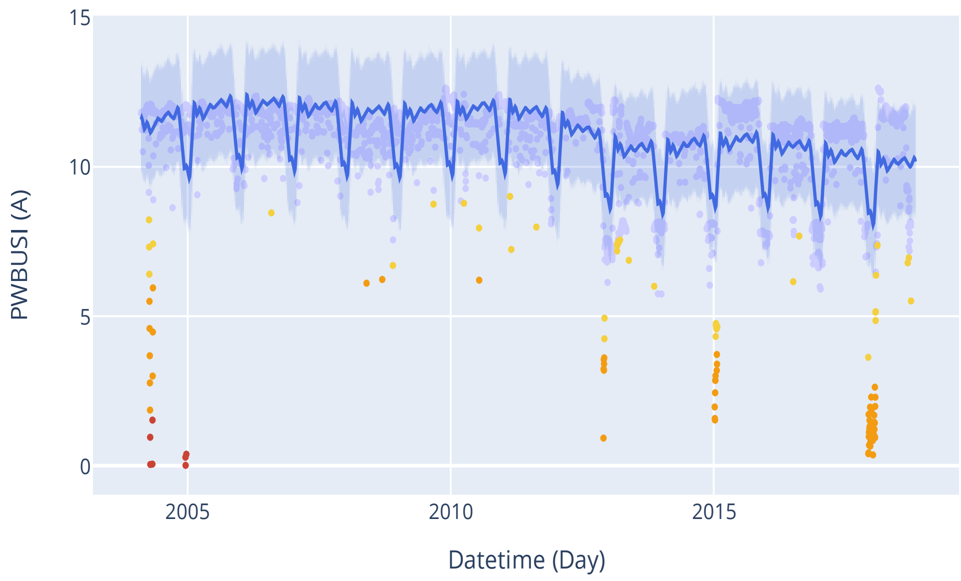

Figure 22 Observed Advanced Anomalies/outliers Spacecraft Bus Current

Bus Voltage

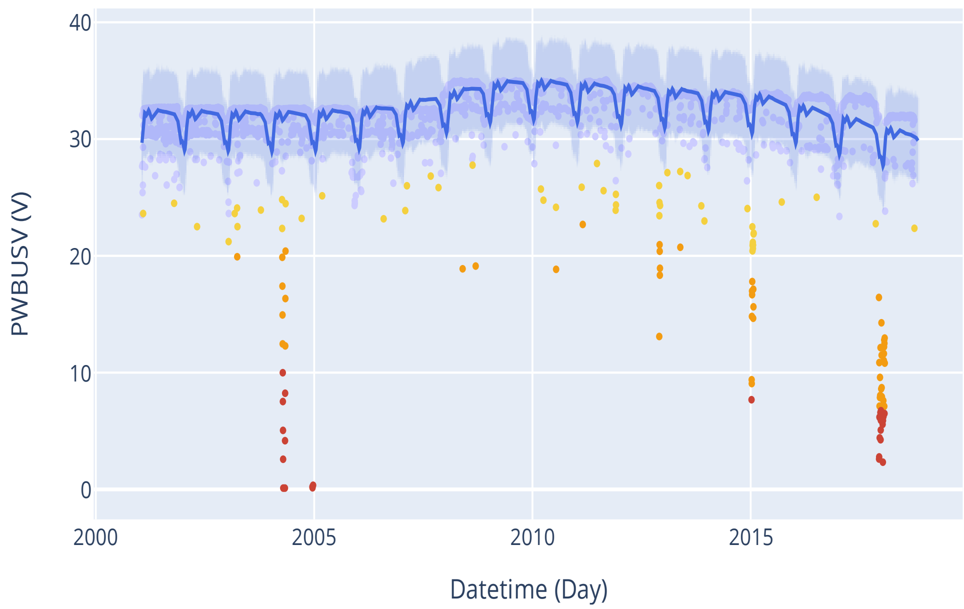

Figure 23 Observed Anomalies/Outliers Bus Voltage

D. Observation

Statistical Model

ARIMA generalizes well in most cases, so it is a powerful model for forecasting. But it takes a lot of computational time, and it needs hyper-parameter tuning that requires predefine the parameters, so it needs supervision and requires high hardware for huge datasets**.**

Deep Learning Models:

LSTM and **Autoencoder** tend to be overfitting and sensitive to outliers.

Prophet Model:

Prophet model is robust to missing data and shifts in the trend, and typically handles outliers well. The final selected model will be the Prophet model, which is not just because it can perfectly predict the data pattern, but it is very fast compared to ARIMA and less sensitive to outliers.

CONCLUSION

Time series analysis is one of the many big computational processes created specifically for longitudinal data analysis over the last thirty years. we can only obtain the operational status and health status of an in-orbit satellite through satellite telemetry data. The pattern mining and extraction of satellite telemetry data are of high significance for automatic judgment and anomaly detection. This paper is devoted to time-series strategies for analyzing satellite telemetry data and detecting outliers. This paper discusses various time-series modeling approaches (ARIMA, Facebook Prophet, Long Short-Term Memory, Auto Encoder). We introduced some statistical approaches and supervised and unsupervised machine learning methods (classification, clustering), to summarize telemetry data and give the most significant information in it for monitoring the health state of various subsystems. Experiments conducted on the time series telemetry of the data set have proven the superiority and effectiveness of the machine learning algorithm for data mining in predicting the expected values.

REFERENCES

[1] Charu C Aggarwal. An introduction to outlier analysis. *Springer New York*, 2017.

[3] Laurent-Béatrice Loubes Jean-Michel Cabon Bertrand Barreyre, Clémentine and Loïc Boussouf. Statistical methods for outlier detection in space telemetries. 2018 SpaceOps Conference, 2018.

[4] Josh Blumenfeld. Getting ready for nisar - and for managing big data using the commercial cloud. *E arthdata*, 2019.

[5] George EP Box, Gwilym M Jenkins, Gregory C Reinsel, and Greta M Ljung. *Time series analysis: forecasting and control*. John Wiley & Sons, 2015.

[6] Yingyi Bu, Tat-Wing Leung, Ada Wai-Chee Fu, Eamonn Keogh, Jian Pei, and Sam Meshkin. Wat: Finding top-k discords in time series database. In *Proceedings of the 2007 SIAM International Conference on Data Mining*, pages 449–454. SIAM, 2007.

[7] Varun Chandola, Arindam Banerjee, and Vipin Kumar. Anomaly detection: A survey. *ACM computing surveys (CSUR)*, 41(3):15, 2009.

[8] J. Contreras, R. Espinola, F. J. Nogales, and A. J. Conejo. Arima models to predict next-day electricity prices. *IEEE Transactions on Power Systems*, 18(3):1014–1020, Aug 2003.

[9] Ada Wai-Chee Fu, Oscar Tat-Wing Leung, Eamonn Keogh, and Jessica Lin. Finding time series discords based on haar transform. In *International Conference on Advanced Data Mining and Applications*, pages 31–41. Springer, 2006.

[10] Ryohei Fujimaki, Takehisa Yairi, and Kazuo Machida. An anomaly detection method for spacecraft using relevance vector learning. In *Pacific-Asia Conference on Knowledge Discovery and Data Mining*, pages 785–790. Springer, 2005.

[11] Ryohei Fujimaki, Takehisa Yairi, and Kazuo Machida. An approach to spacecraft anomaly detection problem using kernel feature space. In *Proceedings of the eleventh ACM SIGKDD international conference on Knowledge discovery in data mining*, pages 401–410. ACM, 2005.

[12] Sudipto Guha, Nina Mishra, Gourav Roy, and Okke Schrijvers. Robust random cut forest based anomaly detection on streams. In *International conference on machine learning*, pages 2712–2721, 2016.

[13] M. Gupta, J. Gao, C. C. Aggarwal, and J. Han. Outlier detection for temporal data: A survey. *IE E E Transactions on Knowledge and Data Engineering*, 26(9):2250–2267, Sep. 2014.

[14] GE Hinton and RR Salakhutdinov. Reducing the dimensionality of data with neural networks. science, 313 (5786): 504–507, 2006.

[15] Kyle Hundman, Valentino Constantinou, Christopher Laporte, Ian Colwell, and Tom Soderstrom. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In *Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining*, pages 387–395. ACM, 2018.

[16] HV Jagadish, Nick Koudas, and S Muthukrishnan. Mining deviants in a time series database. In *V L DB*, volume 99, pages 7–10, 1999.

[19] Y LeCun, Y Bengio, and G Hinton. Deep learning. nature 521 (7553): 436. *Google Scholar*, 2015.

[20] Jessica Lin, Eamonn Keogh, Ada Fu, and Helga Van Herle. Approximations to magic: Finding unusual medical time series. In *18th IEEE Symposium on Computer-Based Medical Systems (CBMS'05)*, pages 329–334. Citeseer, 2005.

[21] Pankaj Malhotra, Anusha Ramakrishnan, Gaurangi Anand, Lovekesh Vig, Puneet Agarwal, and Gautam Shroff. Lstm-based encoder-decoder for multi-sensor anomaly detection. *arXiv preprint arX iv:1607.00148*, 2016.

[22] Corey OMeara, Leonard Schlag, and Martin Wickler. *Applications of Deep Learning Neural Networks to Satellite Telemetry Monitoring*.

[23] Fernando Sanchez, Christopher Pankratz, Douglas M Lindholm, Ransom Christofferson, Darren Osborne, and Thomas Baltzer. Webtcad: A tool for ad-hoc visualization and analysis of telemetry data for multiple missions. In *2018 SpaceOps Conference*, page 2616, 2018.

[24] Jürgen Schmidhuber. Deep learning in neural networks: An overview. *Neural networks*, 61:85–117, 2015.

[25] Pascal Vincent, Hugo Larochelle, Yoshua Bengio, and Pierre-Antoine Manzagol. Extracting and composing robust features with denoising autoencoders. In *Proceedings of the 25th international conference on Machine learning*, pages 1096–1103. ACM, 2008.

[26] Li Wei, Eamonn Keogh, and Xiaopeng Xi. Saxually explicit images: Finding unusual shapes. In *Sixth International Conference on Data Mining (ICDM'06)*, pages 711–720. IEEE, 2006.

[27] A. H. Yaacob, I. K. T. Tan, S. F. Chien, and H. K. Tan. Arima based network anomaly detection. In *2010 Second International Conference on Communication Software and Networks*, pages 205–209, Feb 2010.