PAPER • OPEN ACCESS

Satellite battery sensor values prediction using Bayesian ridge regression models

To cite this article: Mohamed A Galal *et al* 2019 *IOP Conf. Ser.: Mater. Sci. Eng.* **610** 012012

Recent citations

- Spatial Downscaling of Land Surface Temperature Based on a Multi-Factor Geographically Weighted Machine Learning Model
Saiping Xu *et al*

View the article online for updates and enhancements.

Satellite battery sensor values prediction using Bayesian ridge regression models

Mohamed A Galal [1,3]**, Wessam M Hussein**[2] **and Ezz El-din Abdelkawy** [1]

[1]Department of Electrical Engineering, Military Technical College, 11766, Cairo, Egypt

[2]Department of Mechatronic Engineering, Military Technical College, 11766, Cairo, Egypt

[3] E-mail: M_galilio@hotmail.com (Mohamed A Galal)

Abstract: Proper mission control plays a key role in the lifetime of space mission operation, as it ensures that all resources are efficiently utilized when achieving mission goals. Ground control station operation mainly depends on received telemetry together with models simulating spacecraft`s subsystems. Created models help in raising the level of autonomy of MCC (Mission Control Center). Data driven models describe the actual state of the subsystem in real operation situations rather than theoretical costly physical models. This paper proposes data driven models for satellite battery subsystem based on Bayesian ridge regression algorithm. The ridge coefficients minimize a penalized residual sum of squares Thirty models of all thirty battery variables (capacitance, voltage, pressure and temperature) are built from normal operation data. Sensor reading value can be predicted from an observation of all other 29 values. Faults present in sensors or in system can be detected if predicted values are not equal to actual downloaded data from satellite. Bayesian ridge regression models are validated in terms of slope, intercept, R2-value, Q2 -value P-value and standard error.

1. Introduction

A key requirement in designing any space mission is its lifetime. Mission lifetime depends on many key features. One of the most key features in defining space mission lifetime is mission operations, which is how operations are held to carry out the mission objectives. Consequently, Mission Control Center (MCC) has the full responsibility of achieving efficient mission operations. MCC is the only link between satellite and its control to achieve mission objectives. Resources allocation along the lifetime of the spacecraft together with satellite onboard subsystems fault detection and troubleshooting are the main tasks of MCC.[1]

Specialists' level of training and ground software capabilities are the main key roles in efficient mission control. Both aspects integrate each other's. High level of autonomy of ground software can make MCC specialists tasks more easily and efficient. Ground software may contain models simulating space environment, onboard software and onboard subsystems. These models help in increasing MCC level of autonomy as they help MCC specialists to perform their tasks. Created models are mostly physical models, where at the stage of designing and testing the space system, all subsystem designers develop these models. Physical models are very accurate and may describe all subsystems in details, however, they are very costly in terms of money and time. Physical models cannot also relate all variables to each other's when there is no physical or mathematical relation.

Another solution to create models simulating onboard satellite systems is to use data driven model. Such models are especially effective if it is difficult to build knowledge-driven simulation models [2]. Another advantage is the availability of big data coming from ground testing stage as well as daily received telemetry. Besides, Machine learning and artificial intelligence fields have been dramatically developed through past years due to availability of advanced computers that can perform complex and time-consuming computations very quickly and efficiently.

In this paper Bayesian ridge regression models have been developed to predict all onboard satellite battery sensors` readings with high values of R2-value, Q2 -value metrics. Created capacity model has been used as an example to test its capability of fault detection using a test set containing actual battery failure observations together with normal operation observations. Satellite battery normal operation sensor readings have been collected to build the models. Received telemetry of capacitance, voltage, pressure and temperature values of battery have been used to build regression models that predict each value with the knowledge of all others.

2. Mission control center tasks

MCC plans and operates the entire space mission, including the configuration and scheduling of resources for both space and ground system. It computes and issues information needed by other ground system elements and data users. Data concerning the spacecraft's orbit, session times, and antenna pointing angles are submitted to the facilities responsible of antenna operation and payload tasking. MCC tasks include carrying out the detailed mission plan based on the general annual plan. MCC also ensures spacecraft`s health and safety. MCC in general sends commands to the spacecraft during sessions to perform mission tasks, while monitoring its performance. Besides, recovery from off-normal situations is another MCC task. To ensure the designed satellite lifetime operation, proper planning and satellite subsystem analysis tasks must be performed efficiently. Based on proper subsystems analysis, satellite`s resources and health are both well defined, then a proper plan is created and implemented.

For proper resources allocation, all commands coming out from the mission control center must be first verified before sending to spacecraft. Commands verification is a main responsibility of mission control center specialists. Depending on specialists' level of training and ground software capabilities, the task of commands verification complexity varies. Commands verification task must be properly achieved for two main reasons. First reason is to be sure that spacecraft is capable of performing the tasks related to such commands without any faults in onboard subsystems. The second reason is to be sure that satellite resources are efficiently utilized to perform such commands. Level of autonomy of ground software may help MCC specialists to perform such task more easily, as the software itself may reject commands that may cause faults.[1][3]

The task of onboard subsystem fault detection and troubleshooting is also one of the main tasks of MCC. This task varies according to several aspects. For geostationary satellites, where communication with satellite is available all the time, all MCC tasks are real time tasks and telemetry data is always available making it easier to control. However, remote sensing satellites control may be harder task to perform by MCC specialists. Due to limited number of sessions per day, and limited session durations, onboard subsystem analysis for the sake of satellite state of health monitoring is a harder task for MCC specialists to perform in remote sensing satellites, especially when MCC controls a constellation of satellites.

To increase MCC level of autonomy, ground software may contain satellite simulation models. These models may be used to verify commands before sending to satellite. They are also used to automate onboard subsystems health monitoring task. Besides, simulation models may also be used for MCC personnel training.[4]

A traditional, efficient and accurate approach is to develop subsystem models by satellite designers. However, this approach costs a lot in terms of money and time. A comparable approach is to create data driven models with high accuracy to simulate onboard satellite subsystems.

3. Bayesian Ridge regression

[5][6] Having a dataset containing variable (y) denoted as the response that is needed to describe in terms of $(x_1, x_2, \ldots\ldots, x_p)$ variables denoted as the predictors ,then we can predict this variable (y) using linear regression model which can be mathematically presented as follows:

$$\hat{y}(w,x) = w_0 + w_1 x_1 + w_2 x_2 + \ldots.. + w_p \tag{1}$$

where:

$\hat{y}$ is the predicted value

$(w_1, w_2, \ldots.., w_p)$ are the weight coefficients relating X and y.

w_0 is the intercept

Ordinary least square (OLS) regression fits a linear model with coefficients $w = (w_1, w_2, \ldots.., w_p)$ to minimize the residual sum of squares between predicted values and responses present in dataset. This can be mathematically presented as:

$$\min \| Xw - y \|_2^2 \tag{2}$$

To overcome the problem of overfitting, a regularization factor is presented to the loss function to be minimized, such that to be a penalty on the size of coefficients. The ridge coefficients minimize a penalized residual sum of squares as follows:

$$\text{Min} \|xw - y\|_2^2 + \alpha \|w\|_2^2 \tag{3}$$

where α is a complexity parameter that defines the model complexity and in turn model overfitting.

λ is defined as the regularization parameter and it is equal to (α/β) where (β) is inverse the variance of the distribution. As λ increases, the amount of shrinkage increases and the coefficients become more robust to collinearity, leading to less overfitting. small values of λ allow the model to become finely tuned to the noise on each individual data set leading to large variance. Conversely, a large value of λ pulls the weight parameters towards zero leading to large bias. Introduction of the new λ parameter to the loss function changed OLS regression to be called ridge regression.

Instead of using a fixed value for λ, it can be learned from data using Bayesian approach. λ can be treated as a random variable that is estimated from data.

To obtain a fully probabilistic model, y is assumed to be gaussian distributed around Xw, given that:

$$P(y|X, w, \alpha) = N(y|Xw, \alpha) \tag{4}$$

## 4. Sensor reading prediction using Bayesian ridge regression models

Data used to create the models are onboard battery sensor readings of voltage, pressure, temperature and capacitance. Each observation includes 30 readings as follows: 21,4,4,1 voltage, pressure, temperature and capacitance respectively. Available dataset has been split into training and test sets. Test set has been taken to be one third of the whole dataset, where random observations are held. For created models validation, it is needed to validate both goodness of fitting and capability of prediction. First, to describe the properties of the fitting line and validate how much does it really fits data; slope, intercept, R2-value, P-value and standard error parameters are calculated for each created model. Slope and intercept can describe the fitting line properties, while only R2 value can be used to measure the goodness of fitting.
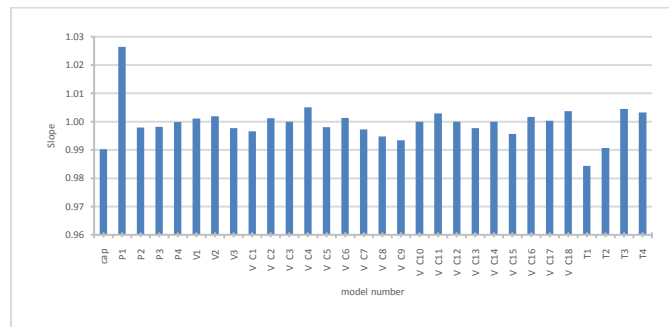
**Figure 1.** slope of each model

By observing figure (1) showing the calculated slope for each linear regression model of all sensors, the model predicting P1 (first sensor pressure) has the largest slope of (1.03), while the model predicting (T1) (first temperature sensor) has the lowest slope of (0.98). However, all slopes have an average approximately equals one (0.99954). This value has no great significance about fitting or prediction, but it only tells about the rate of change of the predicted variable value with respect to other values, which is found to be moderate slope (regression is neither very sensitive nor having poor response).

2 - R2-value (coefficient of determination), representing how close the data are to the fitted regression line, in other words, how much does the line fits data. Values are found to be as in the following figure (2):
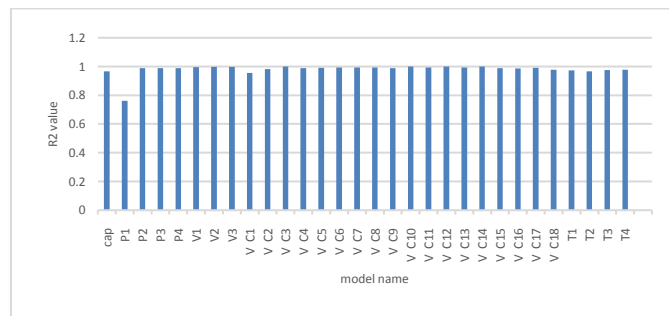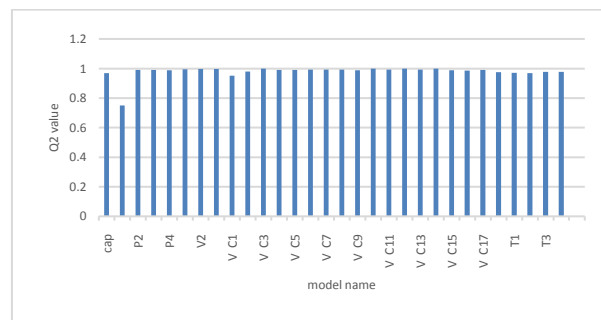

Figure 2. R2-value of each model

All models (except one) have very high R2 values (over 0.96), which means very good fitting of data. The model built to predict P1 (first battery pressure sensor value) has the lowest R-2 value as shown in the above figure. An R-2 value of (0.76) is not high enough, but it still can be used.
Standard error is also a metric that may be used to judge created models` efficiency. As shown in figure (3), model achieved very low values of error.
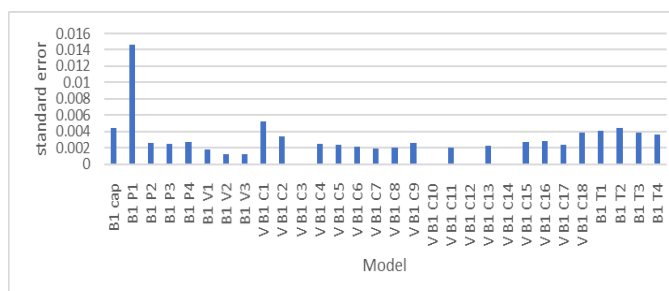
**Figure 3.** standard error of each model

Second, concerning regression prediction power (goodness of prediction), model prediction capability is something different from the model fitness capability, which is the ability of the model to estimate the response for objects that do not participate to the calculated model. Q2 value is a parameter used to validate model prediction power. Q2 is the cross-validated R2, where R2 is calculated by using the same data used to build the model, while Q2 is calculated by using other observations which didn't participate in model building.

R2 and Q2 calculations are almost identical, with the only difference being that residual sum of squares(RSS) is calculated from the data on which the algorithm is trained and Predictive Residual Error sum of squares(PRESS) is calculated from held out data. Q2 for all thirty models are as follows in fig (4)

Figure 4. Q2-value of each model

Q2 values appear to be high as R2 values, except also for model used to predict (P1) first battery pressure sensor.
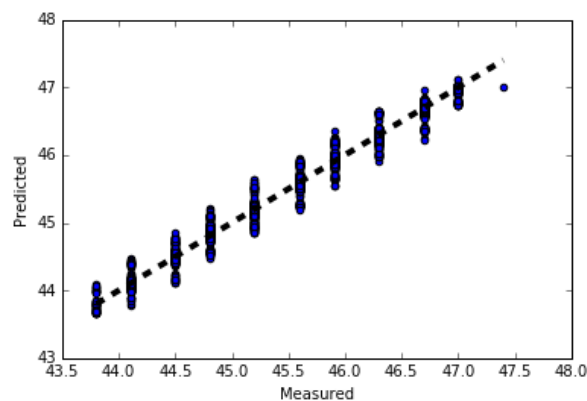

Figure 5. predicted value of pressure sensor vs actual measured value

Figure (5) shows the predicted values of one of the four pressure sensors using the model vs the actual onboard measured value using the test set, and it illustrates that the slope of the line passing through values is almost of 45 degrees, which indicates a good quality of prediction of the model.

5. Models usage in fault detection

Created models can be used for fault detection when predicted values are far from measured values. At this case, either the model fails to predict the accurate actual value, or the value itself is far from normal. When one of the thirty created models has been used to predict another test set comprising of normal observations and new fault observations after system failure which have never been used in models creation, both predicted and measured values were far from each other's only in fault region.

As shown in the bellow figure no. (6), during normal battery operation, the predicted capacity values almost coincided with actual measured values. In fault region, both predicted and measured values are far away. Not only has the model succeeded in anomaly detection in subsystem fault region, but also it succeeded in identifying two other regions of strange behavior. These two regions are when the satellite has been subjected to flight tests, consequently, abnormal battery behavior, as all satellite power coming from solar arrays and battery has been utilized to perform such tests.
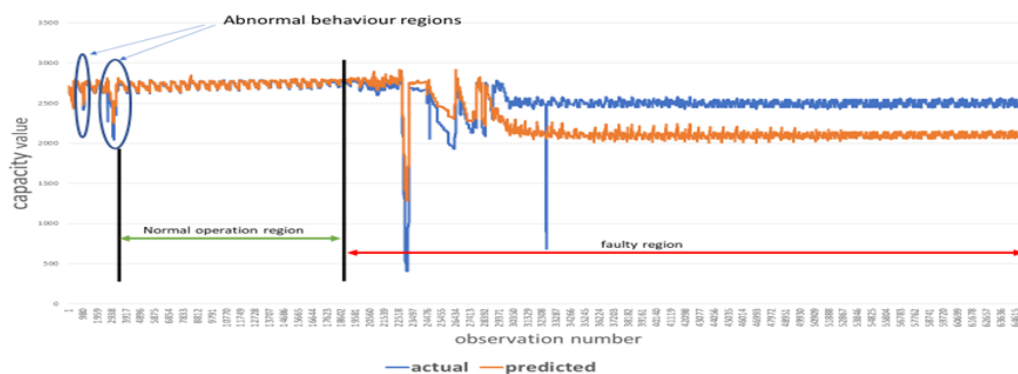
Figure 6. Capacity actual measured values vs predicted using the model, and regions of normal and fault operations

6. Conclusion

In this paper, the two states of the art algorithms for object detection (Faster RCNN and SSD) applied to detect vehicles in satellite images through Transfer Learning and making an experimental analysis comparison between them. We construct vehicle dataset collected by Google Earth and other satellite samples such as JF-2 and WORLD-VIEW satellites. The Inception-V2 used as a base network to enhance the accuracy of detection. Enlarge and increase the variety of training data by using Augmentation techniques. Mean average precision (MAP) used for performance evaluation. Based on the results obtained, Faster R-CNN Inception-V2 gives better accuracy than SSD Inception-V2. but the SSD Inception-V2 performs in a shorter time for image detection. The study will extend for general vehicle detection (bicycle, motorcycle, bus, truck).

References

[1]   Wiley J. Larson (ed.), space mission analysis and design 3rd. ed .

[2]   D. Solomatine, L.M. See and R.J. Abrahart, "Data-Driven Modelling: Concepts, Approaches and   Experiences" ,*Water Science and Technology Library 68, Springer-Verlag Berlin Heidelberg 2008.*

[3]   Daryl G. Boden and Wiley J. Larson , cost effective space mission operations.

[4]   Thomas Uhlig, Florian Sellmaier, Michael Schmidhuber, *Spacecraft Operations,* DOI 10.1007/978-3-7091-1803-0_2).

[5]   Christopher M. Bishop, *pattern recognition and machine learning*, ISBN-10: 0-387-31073-8).

[6]   Radford M. Neal, *Bayesian learning for neural networks*, Springer, ISBN 978-0-387-94724-2)