Reaction Wheels Fault Isolation Onboard 3-Axis Controlled Satellite using Enhanced Random Forest with Multidomain Features

Mofiyinoluwa O. Folami[1], Afshin Rahimi[2]

[1,2] *Department of Mechanical, Automotive, and Materials Engineering,*
University of Windsor, Windsor, ON, N9B 3P4, Canada

folami@uwindsor.ca
arahimi@uwindsor.ca

ABSTRACT

As the number of satellite launches increases each year, it is only natural that an interest in the safety and monitoring of these systems would increase as well. However, as a system becomes more complex, generating a high-fidelity model that accurately describes the system becomes complicated. Therefore, imploring a data-driven method can provide to be more beneficial for such applications. This research proposes a novel approach for data-driven machine learning techniques on the detection and isolation of nonlinear systems, with a case-study for an in-orbit closed loop-controlled satellite with reaction wheels as actuators. High-fidelity models of the 3-axis controlled satellite are employed to generate data for both nominal and faulty conditions of the reaction wheels. The generated simulation data is used as input for the isolation method, after which the data is pre-processed through feature extraction from a temporal, statistical, and spectral domain. The pre-processed features are then fed into various machine learning classifiers. Isolation results are validated with cross-validation, and model parameters are tuned using hyperparameter optimization. To validate the robustness of the proposed method, it is tested on three characterized datasets and three reaction wheel configurations, including standard four-wheel, three-orthogonal, and pyramid. The results prove superior performance isolation accuracy for the system under study compared to previous studies using alternative methods (Rahimi & Saadat, 2019, 2020).

1. INTRODUCTION

With the ever-growing number of satellites being launched into space, it is crucial that the health monitoring and safety of these systems be advanced enough to compensate for the lack of redundant components and their decreasing size. The attitude determination and control subsystem (ADCS), which employs reaction wheels (RWs) as actuators, is one of the most critical components of a satellite subsystem. By accelerating or decelerating the flywheels attached to the electric motor, RWs can correct the satellite's orientation or perform maneuvers under disturbances (Rahimi et al., 2017). Smaller satellites require less cost for design and mass-production, meaning multiple can be launched into space at a time. Therefore, to ensure reliability and mission success, the health and maintenance of these systems are essential. As a result, fault detection and isolation (FDI) methods for ADCS has developed great incentive to be improved and advanced.

The two main categories for FDI approaches are model-based and data-driven. In model-based approaches, the continuous comparison between the actual system state and nominal state is conducted and in turn, generates residuals. This is achieved by using a mathematical model of the system with ideal behavior. One major advantage of this method is that it has the ability to provide a description of the dynamic behavior and physical understanding of the system (Tidriri et al., 2016). However, developing an accurate mathematical model that considers uncertainties and modeling errors is difficult due to some uncertainties being unquantifiable. As previously mentioned, model-based methods compare the available data with prior information and generate residuals. One method of residual generation is through observer-based techniques. If the system is observable (i.e., it is possible to determine the entire system behavior from the outputs) and the process parameters are known, then it is possible to estimate the output of a process with an observer and the residuals. In (Jia et al., 2019), two Non-Linear Observers (NLO) are developed to robustly reconstruct bias faults and effectiveness factors of RWs, and a systematic observer design method is obtained. Fault-tolerant attitude control for the spacecraft system was not considered in this study, however. Four classifiers, Random Forest (RF), SVM, Partial Least Squares (PLS), and Naïve Bayes (NB), are combined to create a fusion framework for FDI in RWs (Nozari et al., 2019). In (Nemati et al., 2019), nonlinear observers are used for fault diagnosis of spacecraft. The observers are designed to ensure convergence of the estimation error to zero for the nominal nonlinear system. Fault isolation is achieved using a General Observer Scheme (GOS).

https://doi.org/10.36001/IJPHM.2021.v12i2.3078

However, when dealing with model-based approaches, there are certain limiting factors such as system complexity, high dimensionality, nonlinearity processing.

Data-driven approaches focus on analyzing system outputs and rely immensely on large amounts of data, allowing these methods to perform well with large-scale and complex systems, as well as reduce time and costs since the development of models is unnecessary. This approach can be beneficial if no mathematical model or expert knowledge about a system is available.

A highly nonlinear dynamic system is the result of modelling the complexity of a single RW. More RWs can be placed into the assembly combined with the attitude and dynamics of the satellite in orbit. If the parameters of interest are non-measurable to the FDI scheme, it becomes difficult to detect faults and isolate their root causes. When conducting data-driven FDI, there are two critical factors any scheme should have, 1) generation of a feature subset, 2) selection of the most optimal classifier. Research conducted by (ElDali & Kumar, 2021) proposed a growing neural network for aircraft engine fault diagnosis, detection, and prognostics. The model optimized the Long-Short Term Memory (LSTM) algorithm and was used to detect failure for RWs in the pyramid configuration. The model tried to detect failures in the four RWs by calculating a Health Index value. A value between 0 and 1, with the latter representing a greater chance of failure. The values were acquired from the residuals of the acquired speed values and the predicted ones. The study (Lee et al., 2020) addresses fault detection and identification in the RWs of nanosatellites based on a Deep Learning (DL) algorithm. Fault detection is accomplished using the residuals between the measured attitude and the estimated attitude. Trial and error were used to optimize the learning and hyperparameters of the LSTM model used within this study. However, this technique did not consider the temporal elements of the attitude information and was more likely to misjudge because only the instantaneous values were used.

Most of the research above also involves instances where residual generation is applied through analytical models. This method, however, does not do well in explaining the facility point behind the detected faults. Mixed learning models are limited to the performance of the best classifier used within the ensemble. If other classifiers have limitations, it can hinder the overall accuracy of the method. Furthermore, most methods deal with only one RW and do not consider multiple faults or different RW assemblies. Based on the open literature of past studies, a research objective was established. Which is the design and development of a data-driven FDI scheme that is capable of autonomously isolating the location of faults within a nonlinear system, with a case-study for an in-orbit closed-loop controlled satellite with reaction wheels as actuators This method is to be developed to be applicable to other systems, however it is being designed for this specific application. To achieve the above research objective, Time Series Feature Extraction Library (TSFEL) transforms the time-domain data into a feature-space state. The new feature space is then fed as input into multiple machine learning models (MLMS). Sensitivity analysis of missing sensors, missing values and noise analysis was also conducted on various datasets. The robustness of this method was also tested on three different configurations of the Reaction Wheel Assemblies (RWA). Lastly, k-fold cross-validation is used to validate the results. The choice for using less computationally expensive methods versus more advanced and highly computational methods was made to ensure applicability and accessibility of the proposed in this study on a broader range of satellite units with lower-end computational capacities. This is particularly important as smaller satellites with limited capacity may not house supercomputers onboard, and computational power and energy consumption can be limiting factors in deploying such algorithms for the FDI of such units.

Past research on data-driven approaches shows that these methods are known to have some limitations. The proposed method can deal with the extensive consumption of resources, memory, and algorithmic complexity by applying a hierarchical approach to enhance the isolation scheme and only utilizing the appropriate resources when required.

The novelties of this approach are:

1. Based on previous assessments, common feature subsets such as statistical features do not result in the best results. Therefore, an automated feature extraction library is used to obtain the best feature subset from a temporal, statistical, and spectral domain which has not been explored for this specific application, to represent the characteristics of the multiple in-phase faults from different points of view and where most literature only consider one space-state transformation.
2. Consideration of three different fault time characteristics: Abrupt, Transient, and a general case. Where most literature mainly considers abrupt faults.
3. Use of variable inception time and duration of the faults for different cases to tackle the applicability range of data-driven methods and broaden the scope, where other literature only considers constants.
4. Classification of multiple in-phase faults where other literature primarily considers either single faults or multiple out-of-phase faults.

The remainder of this paper is organized as follows: Section 2 presents the problem definition. Section 3 explains the methodology employed in the study. Section 4 presents the case studies used to evaluate the proposed method. Section 5 offers insight into the results and discussion of the study. Section 6 concludes the paper with final remarks.

2. PROBLEM DEFINITION

A nonlinear system in discrete-time state space can be represented in Eq. (1)

$$\Omega = \begin{cases} \xi_{k+1} = f(\xi_k, u_k, \theta_k, w_k^{\xi}) \\ \theta_{k+1} = \theta_k + w_k^{\theta} \\ y_k = g(\xi_k, \theta_k) + v_k \end{cases} \tag{1}$$

where at time step k, $\xi_k \in \mathbb{R}^n$ is the state vector, $uk \in \mathbb{R}^m$ depicts the control input vector, $\theta_k \in \mathbb{R}^l$, $y_k \in \mathbb{R}^m$, $w_k^{\xi} \in \mathbb{R}^n$, $w_k^{\theta} \in R^l$ and $vk \in \mathbb{R}^m$ represent the system parameter vector, measurement vector, additive process noise for states, additive process noise for parameters, and additive measurement noise, respectively. $f(\cdot)$ is a nonlinear process model, and $g(\cdot)$ is a nonlinear measurement model. Full state measurement $y_k = \xi_k + v_k$ is considered. In this study, it is under the assumption that the system component faults are reflected as changes in the physical system parameters (Sobhani-Tehrani et al., 2014). Eq. (2) describes the faulty system, also known as a multi-parameter fault model, with:

$$\theta_k = \theta_o + \alpha_k \tag{2}$$

where $\theta_o \in \mathbb{R}^l$ is the nominal parameter vector representation and the fault parameter vector containing L fault elements is represented by $\alpha_k \in \mathbb{R}^L$. By using the fault model given by Eq. (2), one can transform the problem of nonlinear fault diagnosis into the form of an on-line nonlinear parameter tracking problem. Furthermore, for fault isolation, one can extract L single-parameter models Ω_i $i = 1, \ldots, L$. This is shown in Eq. (3) below:

$$\Omega_i \colon \{\theta_k^i = \theta_o^i + \alpha_k^i \qquad i = 1, \ldots, L \tag{3}$$

The mission of the data-driven algorithm is set to classify the current state of the system as one of the possible L faulty cases. In this study, multiple fault scenarios are considered.

3. METHODOLOGY

A hierarchical approach is used within this study for the methodology when tackling the FDI problem. This ensures a reduction in computation time by allocating the right resources to be used only when required. A data-driven fault diagnosis model is created to classify multiple in-phase faults of a satellite RW. The data is simulated using RW data and then extracted for features. The features are calculated using TSFEL and then reduced using Recursive Feature Elimination (RFE) (Bahl et al., 2019) to produce a feature subset ranking. The features are fed as input for the machine learning model (MLM) as inputs for training and testing. The MLMs implemented within this study include Gradient Boosting (GB), Decision Tree (DT), Random Forest (RF), and Multi-Layer Perceptron. The final step includes employing tenfold – cross-validation to provide a more robust estimation of feature selection.

3.1. Data Preprocessing

It requires much time and effort for data science experts to model the Machine Learning Models (MLM) and tune its hyperparameters. For some spacecraft systems, historical data is not always available for training because the data is being collected in real-time. This puts a limitation on the model's time to tune and train the algorithm. It also makes onboard health monitoring a difficult task since the model would have to be created after the data is transmitted to Earth. Auto-machine learning can compensate for the drawbacks of normal machine learning techniques by automating the data pre-processing, model selection, hyperparameter optimization and the interpretation of the results. To assist the MLM in accurately distinguishing between nominal and fault scenarios, the time-series data is transformed into a feature space state. The features are acquired from raw time series data through Python package TSFEL, which extracts over 60 different features across a spectral, temporal, and statistical domain (Barandas et al., 2020). The new representation of the data is what is used as input for the training set. The data is also filtered for missing values, highly correlated features, and low variance features. Finally, the data is normalized before being fed into the MLMs.

3.2. Feature Reduction

Less sensitive and inaccurate features can reduce the model's ability to predict faults accurately. That is why Recursive Feature Elimination Cross-Validation (RFECV) was implemented in the pre-processing stages. RFECV is an iterative backward selection method that uses a classification model to produce a feature subset ranking instead of a feature ranking. All the features are fitted to the model, and at each iteration, the feature with the least importance is eliminated (Ramachandran & Siddique, 2019). For this study, RF is used to train the model, and for each tree, the model estimates the variable importance by recording the out-of-bag prediction accuracy for every predictor variable premutation. The difference between the prior and the altered model accuracy is averaged over all trees and normalized by the standard error at each iteration. The results for RFECV determined that the optimal number of features was 22, as shown in Figure 1.

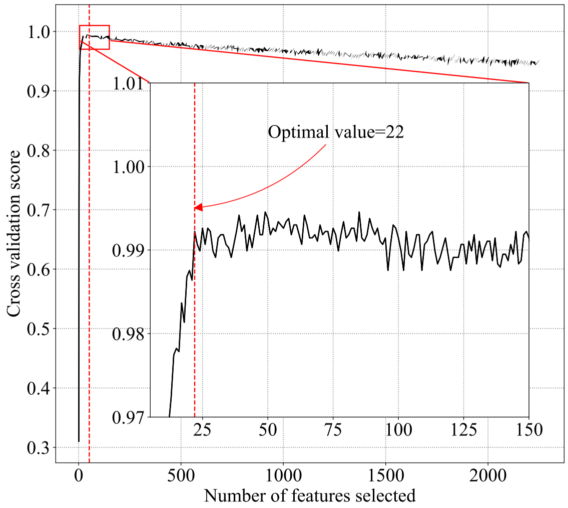


Figure 1. Recursive feature elimination

3.3. Machine Model Selection

In this section, the various machine learning models implemented during the study are briefly discussed. Gradient Boosting (GB) (Bentéjac et al., 2021) is a machine learning technique utilized for regression or classification. This is done by constructing a prediction model comprising an ensemble of weaker models (i.e., Decision Tree). The objective of the GB is to minimize the loss between the actual class value and the predicted class value. The Random Forest (RF) (Probst et al., 2019) classifier consists of many individual decision trees that operate as an ensemble. Each tree will return a class, and the class with the most votes becomes the model's prediction. The low correlation between the trees makes RF a helpful classifier. This protects each tree from errors of other individual trees. A Decision Tree (DT) learns from data to try and approximate a sine curve (Charbuty & Abdulazeez, 2021). This curve has a set of if-then-else decision rules. It breaks down data into smaller subsets while in the same notion incrementally making an associated DT. To enhance the RF and DT classifiers, a meta-classifier AdaBoost is used. It begins by fitting the classifier on the original data set and continues to fit additional copies of the classifier on the original data. However, in successive iterations, the weights of the incorrectly classified instances are adjusted to achieve better classification. The Multi-Layer Perceptron (MLP) is a feed-forward Artificial Neural Network (ANN) (Mishra & Huhtala, 2019). It can be referred to as a network with multiple layers of perceptron (nodes), each with a threshold activation function. There are at least three layers of nodes in an MLP: input layer, hidden layer, and output layer. Besides the input layer, each node uses a non-linear activation function. Backpropagation is used for training. The hyperparameters for the MLP were tuned using GridsearchCV to run each model with different hyperparameters and return the one with the optimal hyperparameters.

3.3.1. Training, Testing, and Validation

The input data was trained and tested at various splits. After a series of runs, an 80:20 train-test split resulted in the best performance for the MLMs. The K-fold cross-validation technique is also implemented to evaluate predictive models by partitioning the original samples into a training set and test set. The benefit of this method is that all observations are used for both training and validation, and each observation is used for validation only once.

In the following section, details on a case-study are provided to evaluate the performance of the proposed methodology.

4. CASE STUDY: FAULT ISOLATION OF SATELLITE RWS

To assess the performance of the proposed FDI method, the faults for RWs onboard a 3-axis controlled satellite are employed. The simulation consists of a high-fidelity nonlinear model of the RW, nonlinear satellite attitude dynamics, and a sliding mode controller (SMC), depicted in Figure 2. It should be noted that in this study, the simulation setup is assumed to have a perfect sensor, and there is no estimator, so the noise system output feeds directly back into the sum junction before the controller. In the following sections, the effect of noise in system measurement is studied under sensitivity analysis discussions for the proposed method. Each component for this case study can be detailed as follows:

4.1.1. Satellite Dynamics

A fully actuated rigid body spacecraft with RWs as actuators under external and internal torques can be expressed as:

$$J\dot{\omega}_{BI}^{B} = -\omega_{BI}^{B} \times (J_s\omega_{BI}^{B} + A_{RW}J_w\Omega) - A\tau_{RW} + \tau_e \qquad (4)$$

where $J_s \in \mathbb{R}^{3x3}$ is the spacecraft's moment of inertia including the RWs and J is defined as $J = J_s - AJ_wA^T, \; J_w \in \mathbb{R}^{4x4} = diag([J_{w1}J_{w2}, J_{w3}, J_{w4}])$ stands for the axial moment of inertia of each RW. the mapping matrix $A \in R^{3x4}$ captures the influence of the actuator's toques to the principal axes of the spacecraft. Angular velocity of the spacecraft relative to the inertial frame expressed in the body frame is denoted by ω_{BI}^{B}. The external torque is represented by $\tau_e \in \mathbb{R}^{3x1}$ and the torque generated by the RW is τ_{RW}. Using quaternions, the kinematic equation for the spacecraft can be formulated as:

$$\begin{bmatrix} \dot{q}_v \\ \dot{q}_4 \end{bmatrix} = \frac{1}{2}\begin{bmatrix} q_4 I + q_v^{\times} \\ -q_v^T \end{bmatrix}\omega_{BL}^{B} \qquad (5)$$

where the unit quaternion is represented by $\begin{bmatrix} q_v \\ q_4 \end{bmatrix}$ with $q_v \in \mathbb{R}^{3x1} = [q_1, q_2, q_3]^T$ and $q_4 \in \mathbb{R}$ representing the Euler parameters that portray the spacecraft body frame orientation with respect to the orbital frame, where $q_v^T q_v + q_4^2 = 1$. The

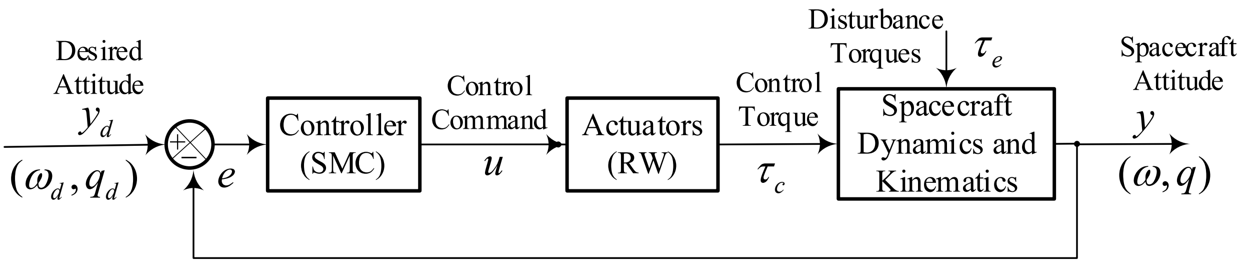


Figure 2. Proposed FDI simulation setup

identity matrix is $I \in \mathbb{R}^{3x3}$ and the skew-symmetric matrix of the quaternion vector $q_v^{\times}$ is expressed as:

$$q_v^{\times} = \begin{bmatrix} 0 & -q_3 & q_2 \\ q_3 & 0 & -q_1 \\ -q_3 & q_1 & 0 \end{bmatrix} \tag{6}$$

4.1.2. Controller

A simplified nonlinear sliding mode controller with quaternion tracking error is defined as (Kumar et al., 2018):

$$\begin{gathered} q_e = q_{d4}q_v - q_4 q_{dv} + q_v^{\times} q_{dv} \\ q_{e4} = q_{d4}q_4 + q_{dv}^T q_v \end{gathered} \tag{7}$$

where $q_{dv} \in R^{3x1}$ and $q_{d4} \in R$ and are the desired attitude, and $q_e^T q_e + q_{4e}^2 = 1$. The rotation matrix is given by:

$$C_e = (q_{4e}^2 - q_e^T q_e)I + 2q_e^T q_e - 2q_{4e}q_e^{\times} \tag{8}$$

Here $C_e^T C_e = 1$, $\dot{C}_e = -\omega_e^{\times} C_e$. Relative angular velocity $\omega_e \in R^{3x1}$ is formulated as follows:

$$\omega_e = \omega_{BL}^B + C_e \omega_d \tag{9}$$

where $\omega_d \in R^{3x1}$ denotes the desired angular velocity. The required control command is defined as:

$$u_r = -\eta A^T \frac{\sigma}{\|\sigma\|}, \ \eta = p_0 + p_1 \|X\| \tag{10}$$

where p_0 and p_1 are positive constants and $X \in R^{6x1} = [q_v, \omega_{BL}^B]^T$. The simplified required input can now be calculated for the actuator given actuator dynamics and required control command as (Kumar, 2018):

$$V_{comm} = R_a K_t^{-1} u_r \tag{11}$$

where $V_{comm} \in R^{4\times1}$ is the voltage input to the RWs, $R_a \in R^{4\times4} = diag([r_{a1}, r_{a2}, r_{a3}, r_{a4}])$ is the armature resistance in ohm and $K_t \in R^{4\times4} = diag([k_{t1}, k_{t2}, k_{t3}, k_{t4}])$ is the motor torque constant for the RWs. All control parameters including λ, p_0 and p_1 are set to the value of 1 for the required simulations in this case study. The full derivation of this controller can be found in (Godard, 2010) on pages 113 to 115.

4.1.3. Actuator

The RW used in this study is an ITHACO "' type A'" by Goodrich. A high fidelity RW nonlinear model was obtained from Bialke (Bialke, 1998) and was integrated into the ACS dynamics. The RW nonlinear model, including discontinuous functions approximated by sinusoidal functions, can be formulated as:

$$\begin{aligned} & I_{RW} = G_d \omega_d [f_3(\omega, I_{RW}) - f_5(\omega)] - \omega_d I_{RW} \\ & + G_d \omega_d V_{comm} \\ & \dot{\omega}_{RW} = \frac{1}{J_w} \{ f_1(\omega) + K_t I_{RW} [f_2(\omega) + 1] - \tau_v \omega \\ & - \tau_c f_4(\omega) + \tau_{noise} \} \end{aligned} \tag{12}$$

where f_1 and f_2 represent the motor disturbance, f_3 denotes the EMF torque-limiting block. The variables f_4, f_5 and V_{comm} represent the analytical approximation of the sign function in the Coulomb friction block, speed limiter block and torque command voltage, respectively.

4.1.4. Reaction Wheel

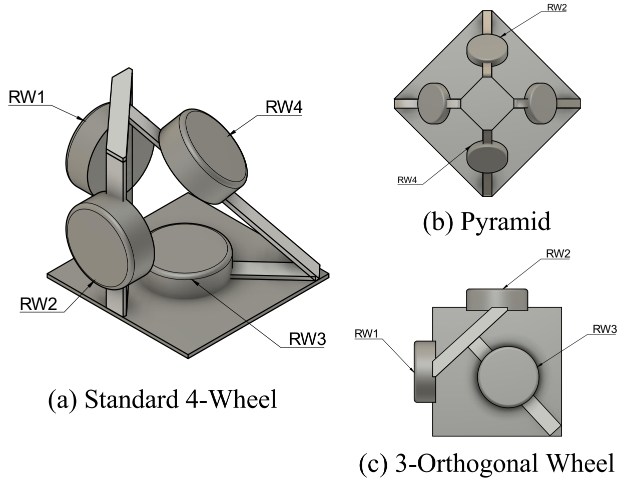
As previously mentioned, the performance of three reaction wheel assembly configurations was studied. The proposed methodology was evaluated on the following configurations: (a) Standard four-wheel with three orthogonal RWs and one redundant oblique RW, (b) Pyramid configuration consisting of four RWs, and (c) three-orthogonal RWs (see Figure 3). The actuator's torque contribution to each principal axis of the spacecraft body frame is mapped as follows (Ismail & Varatharajoo, 2010):

$$\begin{bmatrix} \tau_x \\ \tau_y \\ \tau_z \end{bmatrix} = A_{RW} \begin{bmatrix} \tau_{w1} \\ \tau_{w2} \\ \tau_{w3} \\ \tau_{w4} \end{bmatrix} \tag{13}$$

where τ_x, τ_y, τ_z are the torques applied to the satellite in the x, y, and z direction and τ_{wi} is the torque generated by each RW on its respective axis of rotation. The mapping matrix (A) is as follows:

$$\begin{gathered} A_{RW1} = \begin{bmatrix} 1 & 0 & 0 & -c\beta s\alpha \\ 0 & 1 & 0 & -c\beta s\alpha \\ 0 & 0 & 1 & s\beta \end{bmatrix} \\ A_{RW2} = \begin{bmatrix} c\beta s\alpha & -c\beta c\alpha & -c\beta s\alpha & c\beta s\alpha \\ -c\beta s\alpha & -c\beta c\alpha & c\beta c\alpha & c\beta c\alpha \\ s\beta & s\beta & s\beta & s\beta \end{bmatrix} \\ A_{RW3} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix} \end{gathered} \tag{14}$$

where A_{RWi} is the mapping matrix for configuration i and $i = 1$ refers to the standard 4-wheel configuration, $i = 2$ refers to the pyramid configuration and $i = 3$ refers to the 3-orthogonal configuration. The math functions sine and cosine are abbreviated as c (·) and s (·), and α is the in-plane angle while β is the out-of-plane angle (Kök, 2012).



(a) Standard 4-Wheel

(b) Pyramid

(c) 3-Orthogonal Wheel

Figure 3. Reaction wheel assembly configurations

For the three-orthogonal wheel configuration, a separate dataset had to be generated to account for the proper number of RWs engaged. Cases 1-3 shown in Table 2 referenced in section 4.1.5 are identical to the datasets constructed for the standard and pyramid configurations except for fewer reactions wheels, resulting in a lower number of fault combinations.

4.1.5. Fault Formation

The simulation data is obtained from a closed-loop ACS simulation of a three-axis stabilized low earth orbit (LEO) satellite. Abrupt and transient time-varying fault cases, as well as a generalized fault case, are added into two of the RW components as variations via the bus voltage ($Vbus$) and motor torque back electromotive force (BEMF) constant (kt). $Vbus$ and k_t are modelled as the voltage drop of the power bus and the variations in the torque gain, respectively. The multi-parametrized fault model is obtained by replacing $Vbus,j$ with $Vbus,j0 + \alpha j1$ and kt,j with $kt,j0 + \alpha j2$, where j is the index for the RW unit among the units considered and α_j^i are unknown fault parameters, indicating a possible fault in either the bus voltage or motor current of each wheel. The value for α_j^i will be zero at any given time due to the additive form of the fault parameters. Any deviation from zero for any fault parameter would indicate the fault size and its severity. Within this study, only system satellite outputs, namely q and ω are observed for faulty behavior. Abrupt faults can be viewed as instantaneous faults and can lead to complete failure of the system. Transient faults are more temporary faults within the system and may return to normal parameters given a certain amount of time. This study investigates the effect of fault duration, inception of the fault, and severity of faults during a mission. The initial moment the fault is introduced into the system can be described as the inception time. Duration is the length of time the fault remains within the system. Lastly, the severity denotes the magnitude by which the faulty parameter has deviated from its nominal values during the fault period. With these definitions in mind, three datasets are created when evaluating the proposed methods:

1. **Case 1** (abrupt fault): The inception time for the faults in both $Vbus$ and kt are set at a random value $t_i \in (0,10)$ seconds and the duration of these faults is defined from the inception time to the end of the simulation.
2. **Case 2** (transient bounded fault): Inception time for faults in both $Vbus$ and kt are set at a random value $t_i \in (0,10)$ seconds and the duration of these faults are set randomly within $t_d \in (10,20)$ second range.
3. **Case 3** (transient unbounded fault): The inception time for faults in $Vbus$ and kt are set at a random value $t_i \in (5,55)$ seconds and the duration of each fault is determined randomly within $t_d \in (5,60 - t_i)$ to not exceed the total simulation of 60 seconds.

All cases include a random combination of k_t and V_{bus} faults, with a random severity for each with a set fault deviation range presented in Table 1. The difference in the fault definitions for Cases 2 and 3 lie within the definition of their fault inception time and duration. The larger inception time for Case 3 creates a broader scope or window for fault inception, which is a better representation of real-world scenarios. This distinction for cases 2 and 3 makes them more complex.

Description	Case 1	Case 2	Case 3	Unit
Fault Scenario	0-15/0-7	0-15/0-7	0-15/0-7	integer
k_t in fault	0/1	0/1	0/1	binary
V_{bus} in fault	0/1	0/1	0/1	binary
k_t inception	0-10	0-10	5-55	sec
V_{bus} inception	0-10	0-10	5-55	sec
k_t duration	0-60	10-20	*	sec
V_{bus} duration	0-60	10-20	*	sec
k_t severity	29 ± 2	29 ± 2	29 ± 2	mN·m/A
V_{bus} severity	6 ± 2	6 ± 2	6 ± 2	Volt

Table 1. Fault scenario input format

4.1.6. Dataset Structure

This section provides details of the dataset used for this study.

4.1.6.1 Raw Data

As previously mentioned, it is assumed that the faults injected into the RW units can be detected and isolated by evaluating the satellite output parameters. The satellite attitude parameters include quaternions, q_1 to q_4, and angular speed ω_1 to ω_3. Based on the desired conditions for the case scenarios presented in section 4.5, input values are fed into the MATLAB simulator, which returns the raw data used as input for the FDI algorithm. For this study, 5,000 simulations per scenario were run to create a data set, and since there are 16 scenarios, 80,000 simulations were collected to create one dataset. Since the data pertaining to q_4 is not independent of q_1 to q_3 they were omitted when fed into the FDI algorithm during the pre-processing stage. Of the 80,000 CSV files stored, each one contains a data set of simulation time step of 0.1 seconds, and each simulation has a time length of 60 seconds; therefore, 600 rows are generated. The number of columns is determined as $2 \times 8 = 16$ for two sets of 4 parameters related to the nominal and faulty parameters. This raw data is later transformed into the three different domains through TSFEL, which extracts features from a spectral, statistical, and temporal domain as previously mentioned, and ultimately is fed into the MLMs. Combination theory is used to calculate the total number of combinations for a four-wheel and three-wheel RW assembly, which can be seen in Eq. (15)

$$L = \sum_{p=1}^{n} \binom{n}{p} = \sum_{p=1}^{n} \frac{n!}{p!\,(n-p)!} \tag{15}$$

where p is the number of faulty units at time step k, and n is the number of available units that could become faulty. To be able to refer to each combination (fault scenario), all possible cases are assigned a number in Table 2 for each RW wheel assembly configuration. The fault scenarios for the standard four-wheel and pyramid configurations are the same, while because of the number of wheels involved in the assembly, the scenario numbers and combinations are different for the 3-orthogonal assembly. Once the simulations are executed via the inputs for fault scenarios in Table 1 and the satellite ACS setup in Figure 2, the satellite measurables q and ω are stored in comma-separated value (CVS) files for each simulation separately in the format presented in Table 2 where FW refers to faulty wheels. All the CSV files are used as input for training and testing the MLMs to evaluate their performance under the proposed method and compare their merits.

Configuration	No.	FW	No.	FW
Standard or Pyramid	0	None	8	2,3
	1	1	9	2,4
	2	2	10	3,4
	3	3	11	1,2,3
	4	4	12	1,2,4
	5	1,2	13	1,3,4
	6	1,3	14	2,3,4
	7	1,4	15	1,2,3,4
3-Orthogonal	0	None	4	1,2
	1	1	5	1,3
	2	2	6	2,3
	3	3	7	1,2,3

Table 2. Fault scenarios for RW assemblies

Item	Unit	Description
Time	sec	time in simulation
$q_{i,healthy}$	-	nominal quaternion
$\omega_{i,healthy}$	rad/sec	nominal angular speed for satellite
$I_{RW,ihealthy}$	A	nominal current of RW_i
$\omega_{RWi,healthy}$	rad/sec	nominal angular speed of RW_i
$q_{i,faulty}$	-	faulty quaternion
$\omega_{i,faulty}$	rad/sec	faulty angular speed for satellite
$I_{RWi,faulty}$	A	faulty current of RW_i
$\omega_{RWi,faulty}$	rad/sec	faulty angular speed of RW_i

Table 3. RW simulation outputs

5. RESULTS AND DISCUSSION

The accuracy of classification methods, as discussed in Section 3, which is employed for the FDI problem discussed in Section 2 for the data acquired in the case study detailed in Section 4, is presented in, Table 4, Table 5, and Table 6.

The simulation is tested at a size of 20%. For the cross-validation, the random forest classifier was employed to save on computational costs, the splitting strategy was set to 10-folds, and the parameter that was varied in this analysis was the max depth of the trees.

Method	Case 1 (%)	Case 2 (%)	Case 3 (%)
Gradient Boosting	98.56	80.86	24.64
Random Forest	98.91	69.61	24.36
Decision Tree	92.32	59.14	19.20
MLP	85.64	69.16	21.78

Table 4. Standard 4-wheel configuration accuracy

Method	Case 1 (%)	Case 2 (%)	Case 3 (%)
Gradient Boosting	97.87	51.31	51.65
Random Forest	96.82	54.64	56.05
Decision Tree	93.66	42.19	42.91
MLP	81.42	32.92	32.27

Table 5. Pyramid configuration accuracy

Method	Case 1 (%)	Case 2 (%)	Case 3 (%)
Gradient Boosting	98.02	90.20	36.78
Random Forest	97.65	88.5	35.64
Decision Tree	94.18	83.20	29.77
MLP	94.16	78.2	26.34

Table 6. 3-Orthogonal configuration accuracy

From the results listed in the tables above, the classifier's ability to predict is significantly affected by the fault severity and duration definitions given with the scope of the simulated time series. The case definition described in Section 4.1.5 shows that case 3 is below adequate for any real-world applications. This can be anticipated due to a combination of 1) the fault inception time initiating nearing the end of the simulation time and 2) the fault duration being relatively short. These two problems stem from the influence of the controller. Since the controller compensates for the system when detecting a fault, if the inception time and duration of the fault are not distinct enough, then the system has a hard time classifying the faults. In practical applications of in-orbit satellites, cases 1 and 2 are more practical. If a commanded orientation change causes a fault to occur, the fault will most likely persist to the following command, representing the cases outlined in section 4.1.5. If the MLM fails to identify the fault during a maneuver, the fault will be present in the following command, leading to a true-positive prediction. The learning curves, validation curves, and precision-recall graphs can be seen in Appendix A.

It should also be mentioned that the configuration of the RW assembly influences the accuracy of the proposed method. It is expected that the overall accuracy **of** the cases for the three-orthogonal wheel, standard four-wheel, and pyramid go from best to worst**,** respectively. This is due to the placement of the RWs or the lack of a redundant RW component. Since the three-orthogonal wheel configuration only has three wheels on individual axes, it is not affected by the moments generated by the other RWs. Therefore, if there is a sudden change in the torque generated, there should be a more direct effect on the satellite's behavior. However, this is not the case for the other configurations since the torque generated by these RWs is proportionally changing or contributing to the x, y, and z axes, which results in some direct effect from the three RWs on the x, y and z axes and some indirect contributions from the fourth wheel. The pyramid configuration has a more challenging time with this proposed method due to its assembly's symmetry. If there is an abnormality in the first two RWs and the same discrepancy occurs in the other two RWs the system will have difficulty distinguishing between the two faults.

5.1. Confusion Matrices

The model's performance is evaluated using the test data and used to construct the con-fusion matrices with more details per each scenario. To normalize the results, the number of instances tested per class is used. The results for the confusion matrices are demonstrated in Table 7, Table 8, and Table 8.

Actual	0	146	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	134	0	0	0	0	0	1	0	0	0	0	0	0	0	0
	2	0	0	146	0	0	0	0	0	1	0	0	0	0	0	0	0
	3	0	0	0	148	0	0	0	0	0	0	0	0	0	0	0	0
	4	0	0	0	1	131	0	0	0	0	0	0	0	0	0	0	0
	5	0	0	0	0	0	153	0	0	0	0	0	0	0	0	0	0
	6	0	1	0	0	0	0	150	1	0	0	0	0	0	0	0	0
	7	0	0	0	0	0	0	0	144	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	144	0	0	0	0	0	0	0
	9	0	0	0	0	0	0	0	0	2	146	0	0	0	0	4	0
	10	0	0	0	6	0	0	0	0	0	0	157	0	0	0	0	0
	11	0	0	0	0	0	0	0	0	0	0	0	164	0	0	0	2
	12	0	0	0	0	0	0	0	0	0	0	0	1	149	0	0	0
	13	0	0	0	0	0	0	1	0	0	0	0	0	0	152	0	0
	14	0	0	0	0	0	0	0	0	0	1	0	0	0	0	152	0
	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	162
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		Predicted															

Table 7. Confusion matrix for standard 4-wheel

Along the diagonal, the values that depict the percentage of instances correctly predicted can be seen. Scenario 0 refers to the conditions where all RW units are healthy, and the 100% accuracy establishes the learner's ability to distinguish between nominal and faulty conditions. Scenario 0-4 represents the single faults of the RW where only one unit can fail. Within these sections of the confusion matrices shown above, there is at most only one misclassification amongst the classes. This provides great confidence in the methodology to identify single faults at the very least correctly. The remainder of the scenarios pertain to the definitions described in Table 2. As more units fail simultaneously, the performance of the classifier slightly degrades. This can be explained due to the impact of multiple faulty units on the overall system behavior. As the predictions are made based on the system-level measurements (i.e., satellite quaternions and angular velocities), it is expected for the predictors to identify similar behavior as contributed to by similar actuator faults. Therefore, in any fault scenario, the system-level impact from actuator faults is similar, the predictors will identify them as the same scenario and this can become more evident as more actuator units become faculty simultaneously.

Actual	0	105	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	1	0	94	0	0	0	0	0	0	0	0	0	0	0	0	0	0
	2	0	0	90	0	0	3	0	0	0	0	0	0	0	0	0	0
	3	0	0	0	91	0	0	2	0	0	0	0	0	0	0	0	0
	4	0	0	0	0	90	0	0	8	0	0	0	0	0	0	0	0
	5	0	0	3	0	0	93	0	0	0	0	0	0	0	0	0	0
	6	0	0	0	2	0	0	95	0	0	0	0	0	0	0	0	0
	7	0	0	0	0	4	0	0	94	0	0	0	0	0	0	0	0
	8	0	0	0	0	0	0	0	0	88	0	0	6	0	0	0	0
	9	0	0	0	0	0	0	0	0	0	88	0	0	2	0	0	0
	10	0	0	0	0	0	0	0	0	0	0	110	0	0	0	0	0
	11	0	0	0	0	0	0	0	0	1	0	0	89	0	0	0	0
	12	0	0	1	0	0	0	0	0	0	8	0	0	112	0	0	0
	13	0	0	0	0	0	0	0	0	0	0	8	0	0	89	0	0
	14	0	0	0	0	0	0	0	0	0	0	0	0	0	0	99	5
	15	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1	108
		0	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
		Predicted															

Table 8. Confusion matrix for pyramid

Actual	0	107	0	0	0	0	0	0	0
	1	0	92	0	0	6	0	0	0
	2	0	0	79	0	0	0	0	0
	3	0	0	0	98	0	0	3	0
	4	0	2	0	0	94	0	0	1
	5	0	0	0	0	0	91	0	5
	6	0	0	0	0	0	0	91	0
	7	0	0	0	0	0	9	0	87
		0	1	2	3	4	5	6	7
		Predicted							

Table 9. Confusion matrix for 3-axis orthogonal

5.2. Sensitivity Analysis

Although many MLMs were tested against the proposed method to simplify analysis results, only one machine learning classifier was used for sensitivity analysis. The model's sensitivity is evaluated for missing values, missing sensors, and noise.

5.2.1. Noise Analysis

To study the effects of noisy raw data on the model's performance, Gaussian noise was added at multiple signal-to-noise ratios. Gaussian noise with a zero mean was added during the study. Table 10, Table 11, and Table 12 show the results for the different levels of SNR on the GB classifier. Through the results, it is seen that the model's performance decreases as the SNR decreases. The model contains a reasonable accuracy above 40 dB for the abrupt fault case, which is of greater importance to practical applications.

Standard 4-Wheel							
SNR (dB)	**60**	**50**	**40**	**30**	**20**	**10**	**None**
Score (%)	99.2	98.8	98.4	91.3	75.1	54.7	99.2
3-Orthogonal Wheel							
SNR (dB)	**60**	**50**	**40**	**30**	**20**	**10**	**None**
Score (%)	94.7	93.3	92.3	87.4	76.3	58.2	97.6
Pyramid							
SNR (dB)	**60**	**50**	**40**	**30**	**20**	**10**	**None**
Score (%)	97.3	97.0	96.7	90.8	72.3	48.6	96.8

Table 10. Noise analysis for Case 1 (abrupt fault case)

Standard 4-Wheel							
SNR (dB)	**60**	**50**	**40**	**30**	**20**	**10**	**None**
Score (%)	76.1	71.4	67.0	57.2	43.1	29.3	69.6
3-Orthogonal Wheel							
SNR (dB)	**60**	**50**	**40**	**30**	**20**	**10**	**None**
Score (%)	68.3	70.4	67.2	63.7	53.7	39.9	88.5
Pyramid							
SNR (dB)	**60**	**50**	**40**	**30**	**20**	**10**	**None**
Score (%)	53.3	50.8	49.8	43.8	32.1	21.6	54.6

Table 11. Noise analysis for Case 2 (transient fault case)

Standard 4-Wheel							
SNR (dB)	**60**	**50**	**40**	**30**	**20**	**10**	**None**
Score (%)	19.6	19.5	19.3	15.5	14.8	12.4	24.3
3-Orthogonal Wheel							
SNR (dB)	**60**	**50**	**40**	**30**	**20**	**10**	**None**
Score (%)	38.7	36.4	34.2	34.1	32.1	26.8	35.6
Pyramid							
SNR (dB)	**60**	**50**	**40**	**30**	**20**	**10**	**None**
Score (%)	52.9	51.8	50.1	43.8	31.6	22.9	56.0

Table 12. Noise analysis for Case 3 (general fault case)

5.2.2. Missing Sensor Analysis

For this missing sensor analysis, the fault isolation method was tested under the assumption that not all sensors provide full measurement during the system's operation. Therefore, measurements from some of the sensors are missing due to one or multiple failed sensors. Sensor readings are represented by the satellite quaternions (q_1, q_2, q_3) and angular velocities ($\omega_1, \omega_2, \omega_3$) and the results for the analysis can be seen in Table 13, Table 14, and Table 15, respectively. In each of these tables, various combinations of working sensors (available measurements) are provided along with the isolation method accuracy for the various severity cases introduced in Section 4.1.5 (i.e., Cases 1, 2, and 3).

Available Sensors	**Case 1**	**Case 2**	**Case 3**
	Score (%)	Score (%)	Score (%)
$q_2, q_3, q_4, \omega_1, \omega_2, \omega_3$	99.04	73.40	11.84
$q_1, q_3, q_4, \omega_1, \omega_2, \omega_3$	99.16	70.44	11.47
$q_1, q_2, q_4, \omega_1, \omega_2, \omega_3$	97.96	73.78	12.90
$q_1, q_2, q_3, \omega_1, \omega_2, \omega_3$	98.79	73.28	13.90
$q_1, q_2, q_3, q_4, \omega_2, \omega_3$	98.79	73.28	13.52
$q_1, q_2, q_3, q_4, \omega_1, \omega_3$	98.21	72.13	13.65
$q_1, q_2, q_3, q_4, \omega_1, \omega_2$	97.50	70.57	12.78
$q_1, \omega_1, \omega_2, \omega_3$	98.00	71.03	12.03
$q_2, \omega_1, \omega_2, \omega_3$	97.50	73.34	13.59
$q_3, \omega_1, \omega_2, \omega_3$	99.25	69.66	12.40
q_1, q_2, q_3, q_4	96.58	73.47	13.27
q_1, q_3, q_4	95.46	61.95	10.85
q_1, q_2, q_4	86.56	69.06	10.28
q_1, q_2, q_3	96.67	73.62	13.34
$\omega_1, \omega_2, \omega_3$	98.16	70.75	11.28

Table 13. Standard 4-Wheel missing sensor analysis

Available Sensors	**Case 1**	**Case 2**	**Case 3**
	Score (%)	Score (%)	Score (%)
$q_1, q_2, q_3, \omega_1, \omega_2, \omega_3$	92.82	71.84	26.05
$q_2, q_3, \omega_1, \omega_2, \omega_3$	94.59	71.78	27.55
$q_1, q_2, \omega_1, \omega_2, \omega_3$	93.94	69.53	23.19
$q_1, q_3, \omega_1, \omega_2, \omega_3$	90.91	69.72	23.56
$q_2, q_3, \omega_1, \omega_2, \omega_3$	92.82	71.84	27.55
$q_1, q_2, q_3, \omega_2, \omega_3$	92.38	71.16	26.30
$q_1, q_2, q_3, \omega_1, \omega_3$	91.07	72.40	28.42
$q_1, q_2, q_3, \omega_1, \omega_2$	90.44	71.59	24.93
$q_1, \omega_1, \omega_2, \omega_3$	89.26	66.66	24.06
$q_2, \omega_1, \omega_2, \omega_3$	87.57	69.10	23.56
$q_3, \omega_1, \omega_2, \omega_3$	93.00	68.72	23.19
q_1, q_2, q_3	89.26	72.40	27.30
q_1, q_2	84.78	60.73	21.94
q_1, q_3	73.15	64.10	24.06
q_2, q_3	80.96	65.60	16.58
$\omega_1, \omega_2, \omega_3$	88.01	67.97	26.69

Table 14. 3-Orthogonal missing sensor analysis

As can be seen from the results in Table 13, Table 14, and Table 15, when more measurements are available, the isolation accuracy is higher. As the number of failed sensors increases and there are fewer available measurements for the isolation method to work with, the accuracy decreases. Additionally, it can be observed that the accuracy of the isolation method decreases with the increase in case complexity (Cases 2 and 3 compared to Case 1). This can be attributed to the impact of multiple units on the same measurement for the satellite at the system level. The same trend is observed in the following sensitivity analyses as the accuracy degrades with an increase in the complexity of the testing conditions.

Available Sensors	**Case 1**	**Case 2**	**Case 3**
	Score (%)	Score (%)	Score (%)
$q_2, q_3, q_4, \omega_1, \omega_2, \omega_3$	97.78	54.33	52.68
$q_1, q_3, q_4, \omega_1, \omega_2, \omega_3$	94.70	42.64	43.51
$q_1, q_2, q_4, \omega_1, \omega_2, \omega_3$	96.32	48.44	53.17
$q_1, q_2, q_3, \omega_1, \omega_2, \omega_3$	96.88	50.37	54.05
$q_1, q_2, q_3, q_4, \omega_2, \omega_3$	95.88	48.87	50.12
$q_1, q_2, q_3, q_4, \omega_1, \omega_3$	96.07	48.06	50.74
$q_1, q_2, q_3, q_4, \omega_1, \omega_2$	96.00	49.12	51.30
$q_1, \omega_1, \omega_2, \omega_3$	95.07	44.26	44.95
$q_2, \omega_1, \omega_2, \omega_3$	97.13	50.99	53.11
$q_3, \omega_1, \omega_2, \omega_3$	95.01	43.39	45.07
q_1, q_2, q_3, q_4	93.70	46.25	49.06
q_1, q_3, q_4	69.70	22.81	22.75
q_1, q_2, q_4	92.76	44.38	48.19
q_1, q_2, q_3	94.45	46.57	50.49
$\omega_1, \omega_2, \omega_3$	95.19	45.26	46.19

Table 15. Pyramid missing sensor analysis

5.2.3. Missing Value Analysis

Sensory data containing missing values are not uncommon due to miscommunication between channels or sensor components. In this section, a complete analysis of the model's performance with missing values is evaluated. The simulated data does not initially contain any missing values. Therefore, missing values had to be generated manually at different percentages. Sklearn's Iterative Imputer was used for interpolation during the data pre-processing stage to ensure the proper calculation of missing values. To accomplish this, the iterative imputer modelled each feature with missing values as a function of other features using the round-robin method. To initialize the missing values, the mean along each column was used to replace the missing values. The results for the missing value analysis are presented in Table 16, Table 17, and Table 18, where MMP refers to missing measurement percentage ranging from 0% for full measurements availability and up to 50% of missing measurements in the available data for the isolation method.

Standard 4-wheel configuration									
MMP (%)	**0**	**1**	**3**	**5**	**7**	**10**	**20**	**35**	**50**
Score (%)	99.3	98.1	96.9	95.5	94.3	92.4	87.5	80.2	74.3
3-Orthogonal configuration									
MMP (%)	**0**	**1**	**3**	**5**	**7**	**10**	**20**	**35**	**50**
Score (%)	96.8	92.8	91.8	91.6	92.1	90.7	88.7	84.3	78.1
Pyramid configuration									
MMP (%)	**0**	**1**	**3**	**5**	**7**	**10**	**20**	**35**	**50**
Score (%)	97.7	95.9	95.1	94.0	93.0	93.1	89.7	84.9	80.7

Table 16. Missing value analysis for Case 1 (abrupt fault)

As can be seen from Table 16, Table 17, and Table 18, the accuracy of the isolation method decreases with the increase in the number of missing measurements.

Standard 4-wheel configuration									
MMP (%)	**0**	**1**	**3**	**5**	**7**	**10**	**20**	**35**	**50**
Score (%)	69.6	64.2	61.3	59.2	58.5	59.0	52.8	49.6	44.4
3-Orthogonal configuration									
MMP (%)	**0**	**1**	**3**	**5**	**7**	**10**	**20**	**35**	**50**
Score (%)	88.5	54.1	50.9	50.1	49.1	47.0	45.4	41.6	37.1
Pyramid configuration									
MMP (%)	**0**	**1**	**3**	**5**	**7**	**10**	**20**	**35**	**50**
Score (%)	54.6	50.8	47.3	47.8	47.5	46.8	45.8	45.8	42.6

Table 17. Missing value analysis for Case 2 (transient fault)

Standard 4-wheel configuration									
MMP (%)	**0**	**1**	**3**	**5**	**7**	**10**	**20**	**35**	**50**
Score (%)	24.4	18.7	18.5	18.5	17.3	18.4	16.8	15.2	14.7
3-Orthogonal configuration									
MMP (%)	**0**	**1**	**3**	**5**	**7**	**10**	**20**	**35**	**50**
Score (%)	36.5	31.8	34.7	34.3	33.2	31.9	30.4	27.6	25.8
Pyramid configuration									
MMP (%)	**0**	**1**	**3**	**5**	**7**	**10**	**20**	**35**	**50**
Score (%)	56.1	48.3	47.8	45.4	44.6	44.3	42.3	38.9	35.9

Table 18. Missing value analysis for Case 3 (general fault)

Additionally, it can be observed that the accuracy of the isolation method decreases with the increase in complexity of the testing case (Cases 1, 2, and 3), where Case 1 is the least complex case, and Case 3 is the most complex case.

5.3. Validation and Learning Curve

As previously mentioned in Chapter 3, validation testing was conducted on the methodology. It is sometimes useful to plot the influences of a single hyperparameter instead of selecting multiple to find out whether the estimator selected is overfitting or underfitting hyperparameter values. Based on the training and validation score plot, if both are low, the estimator is underfitting the data. If the training score remains high and the validation score is low, the estimator is overfitting the data. Otherwise, if both are high, then the estimator is working well. Due to the random forest classifier being computational less expensive and having good accuracy results for the more practical case study scenario. It was selected as the estimator to plot the validation curves shown in Figure 4, Figure 5, and Figure 6. The learning curve shows the training score and validation of an estimator for various training samples. This tool examines whether it is beneficial to add more training data and whether the estimator suffers from a bias error or a variance error. The learning curves are shown in Figure 7, Figure 8, and Figure 9.

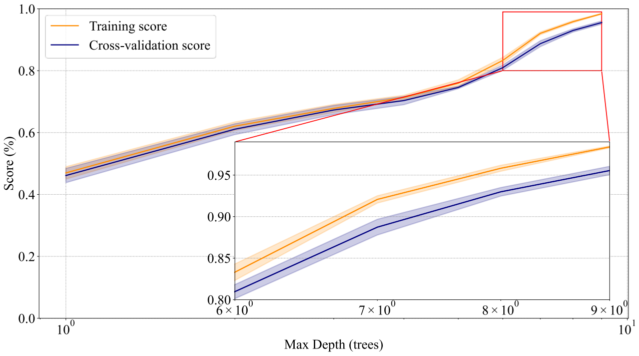


Figure 4. Validation curves for standard 4-wheel

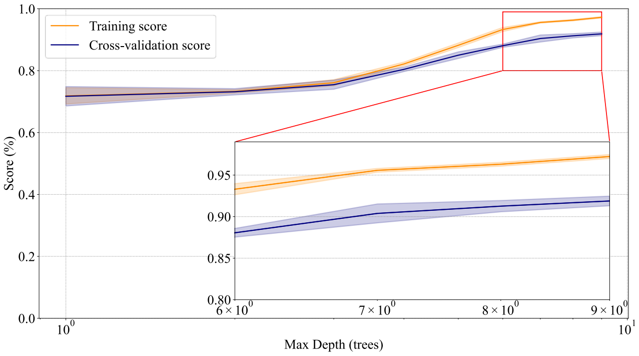


Figure 5. Validation curve for 3-orthogonal wheel

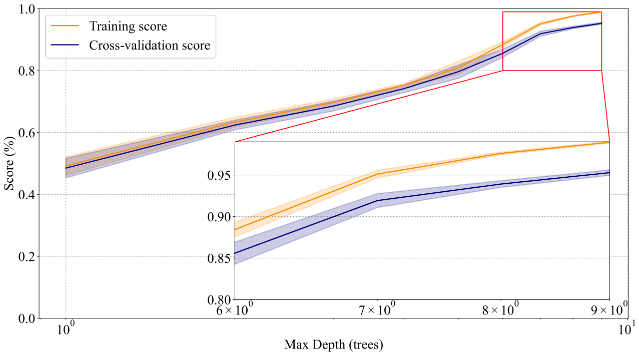


Figure 6. Validation curve for pyramid

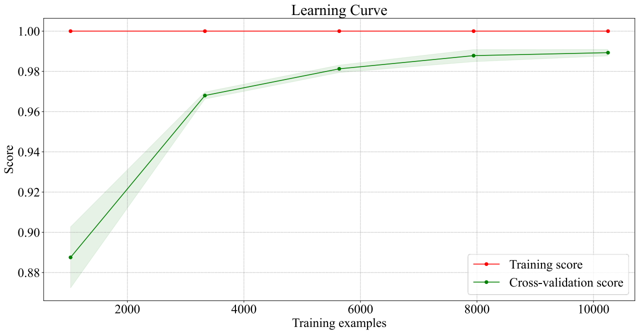


Figure 7. Learning curve for standard 4-wheel

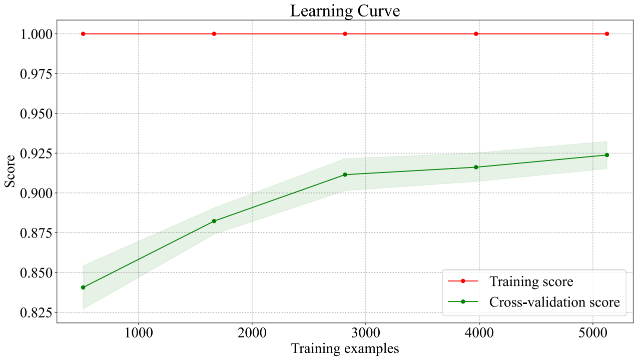


Figure 8. Learning curve for 3-orthogonal wheel

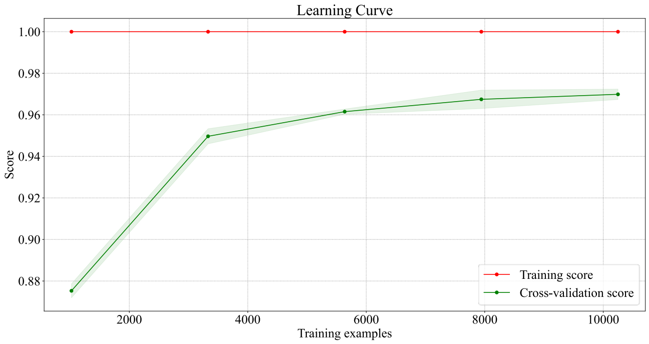


Figure 9. Learning curve for pyramid

5.4. Precision and Recall Curve

The precision and recall curve can summarize the tradeoff between the true positive rate and positive predictive value for a predictive model using various probability thresholds. Precision can be defined as a ratio of the number of true positives divided by the sum of the true positives and false positives and describes how well a model predicts positive classes using

$$Precision = \frac{True\ Positives}{True\ Positives + False\ Positives} \qquad (16)$$

The recall is the ratio of true positives divided by the sum of the true positive and false negatives. It is the same as sensitivity as

$$Recall = \frac{True\ Positives}{True\ Positives + False\ Negatives} \qquad (17)$$

A high area value under the curve represents high precision and recall, where a high precision translates to a low false positive rate, and a high recall relates to a low false negative rate. Thus, high results determine that the classifier is returning accurate results. The precision and recall curves for the three configurations can be seen in Figure 10, Figure 11, and Figure 12. These figures show the precision-recall curves for each class (fault scenario) that was tested under this study.

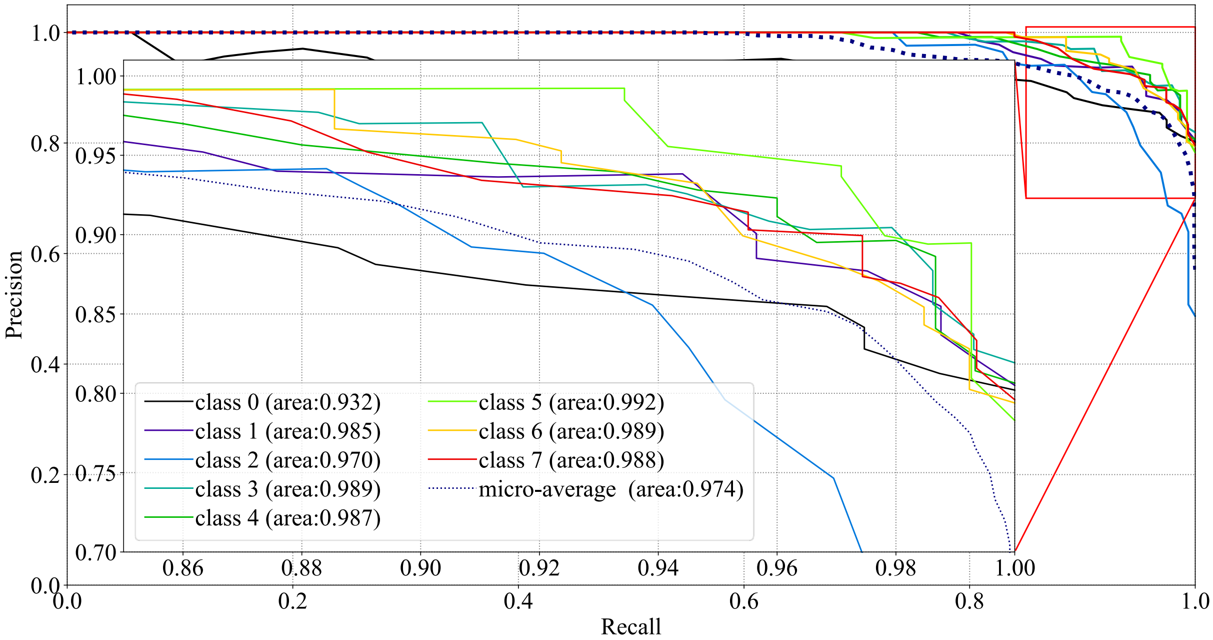


Figure 10. Precision and recall for 3-orthogonal wheel configuration

6. CONCLUSION

In this paper, the development of a data-driven fault isolation machine learning algorithm for nonlinear systems was investigated. The performance of the proposed method was evaluated for a case study involving FDI of an ADCS system using RWs as actuators onboard an in-orbit satellite. An automated feature extraction approach was implemented to transform the data into three different domains, after which the feature reduction was utilized to find the optimal number of features for training. The proposed fault isolation scheme was evaluated against various MLMs and with a thorough sensitivity analysis. The Random Forest classifier resulted in the overall highest accuracy achieved amongst the more applicable cases and the three configurations considered in this study. This provides enough confidence that the proposed method could be a successful candidate in such applications. The future scope of this work includes implementing a change point

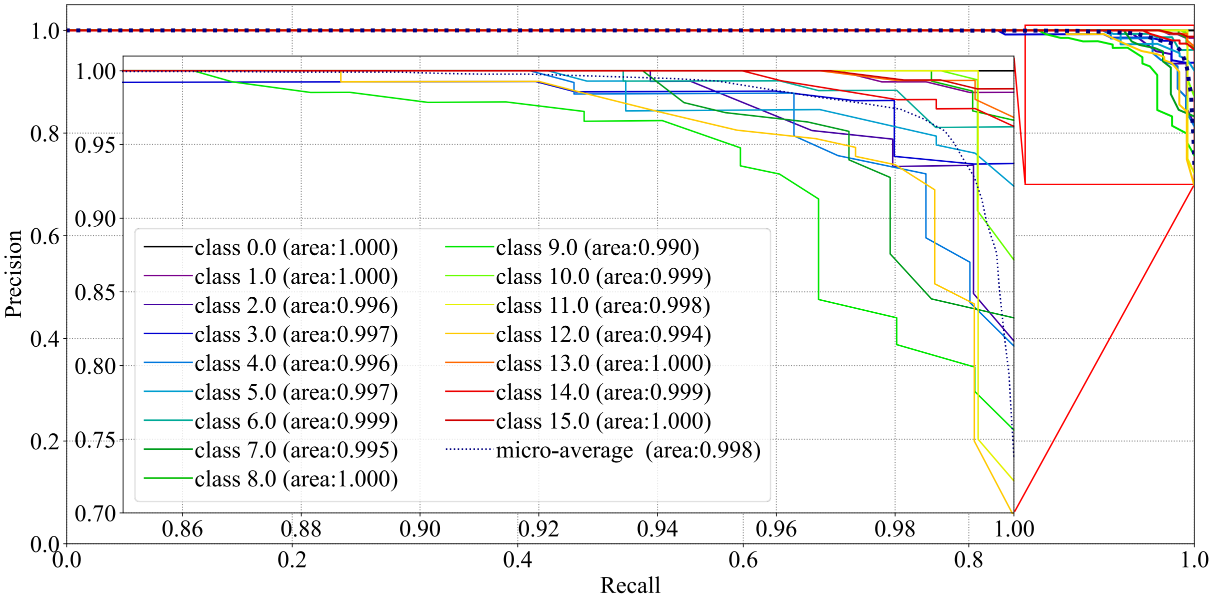


Figure 11. Precision and recall for standard 4-wheel configuration

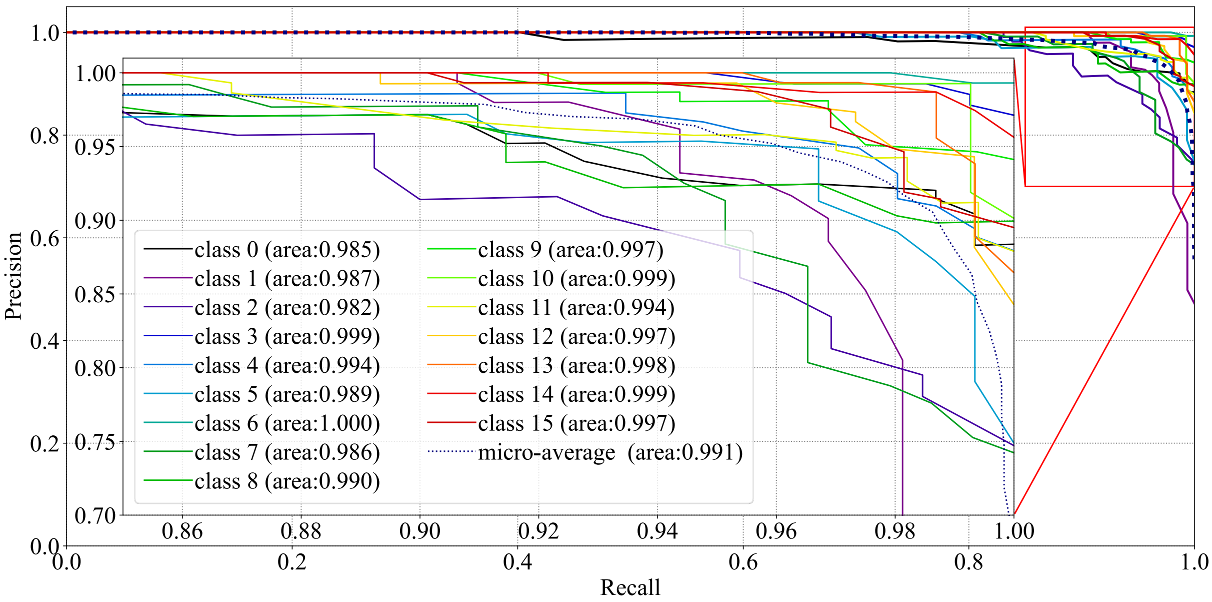


Figure 12. Precision and recall for pyramid configuration

detection algorithm on the more complex testing conditions, Case 3, to help better present any underlying patterns in the dataset. Expanding upon dataset cases is also essential, as this study has determined that the parameters used to develop the faulty dataset significantly impact the ability of the machine learning algorithm to detect a faulty condition correctly. Additionally, empirical data from in-orbit satellites can further enhance the applicability of the proposed method for real-life applications.

Acknowledgement

The authors are grateful to the University of Windsor, and the Natural Sciences and Engineering Research Council of Canada (NSERC) for their financial support.

References

Bahl, A., Hellack, B., Balas, M., Dinischiotu, A., Wiemann, M., Brinkmann, J., Luch, A., Renard, B. Y., & Haase, A. (2019). Recursive feature elimination in random forest classification supports nanomaterial grouping. *NanoImpact*, *15*(May), 100179. https://doi.org/10.1016/j.impact.2019.100179

Barandas, M., Folgado, D., Fernandes, L., Santos, S., Abreu, M., Bota, P., Liu, H., Schultz, T., & Gamboa, H. (2020). TSFEL: Time Series Feature Extraction Library. *SoftwareX*. https://doi.org/10.1016/j.softx.2020.100456

Bentéjac, C., Csörgő, A., & Martínez-Muñoz, G. (2021). A comparative analysis of gradient boosting algorithms. In *Artificial Intelligence Review* (Vol. 54, Issue 3). Springer Netherlands. https://doi.org/10.1007/s10462-020-09896-5

Bialke, B. (1998). High Fidelity Mathematical Modeling of Reaction Wheel Performance. *1998 Annual AAS Rocky Mountain Guidance and Control Conference, Advances in the Astronautical Sciences*, 483–496.

Charbuty, B., & Abdulazeez, A. (2021). Classification Based on Decision Tree Algorithm for Machine Learning. *Journal of Applied Science and Technology Trends*, *2*(01), 20–28. https://doi.org/10.38094/jastt20165

ElDali, M., & Kumar, K. D. (2021). *Fault Diagnosis and Prognosis of Aerospace Systems Using Growing Recurrent Neural Networks and LSTM*. 1–20. https://doi.org/10.1109/aero50100.2021.9438432

Godard. (2010). *Fault Tolerant Control of Spacecraft* [Ryerson University]. https://digital.library.ryerson.ca/islandora/object/RULA%3A1518/datastream/OBJ/download/Fault_Tolerant_Control_Of_Spacecraft.pdf

Ismail, Z., & Varatharajoo, R. (2010). A study of reaction wheel configurations for a 3-axis satellite attitude control. *Advances in Space Research*, *45*(6), 750–759. https://doi.org/10.1016/j.asr.2009.11.004

Jia, Q., Li, H., Chen, X., & Zhang, Y. (2019). Observer-based reaction wheel fault reconstruction for spacecraft attitude control systems. *Aircraft Engineering and Aerospace Technology*, *91*(10), 1268–1277. https://doi.org/10.1108/AEAT-07-2018-0203

Kök, I. (2012). *Comparison and Analysis of Attitude Control Systems of a Satellite Using Reaction Wheel Actuators*. 17–18.

Kumar, K. D., Godard, Abreu, N., & Sinha, M. (2018). Fault-tolerant attitude control of miniature satellites using reaction wheels. *Acta Astronautica*, *151*(May), 206–216. https://doi.org/10.1016/j.actaastro.2018.05.004

Lee, K. H., Lim, S. M., Cho, D. H., & Kim, H. D. (2020).

Development of Fault Detection and Identification Algorithm Using Deep learning for Nanosatellite Attitude Control System. *International Journal of Aeronautical and Space Sciences*, *21*(2), 576–585. https://doi.org/10.1007/s42405-019-00235-9

Mishra, K. M., & Huhtala, K. J. (2019). Fault Detection of Elevator Systems Using Multilayer Perceptron Neural Network. *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, *2019-Septe*, 904–909. https://doi.org/10.1109/ETFA.2019.8869230

Nemati, F., Safavi Hamami, S. M., & Zemouche, A. (2019). A nonlinear observer-based approach to fault detection, isolation and estimation for satellite formation flight application. *Automatica*, *107*, 474–482. https://doi.org/10.1016/j.automatica.2019.06.007

Nozari, H. A., Castaldi, P., Banadaki, H. D., & Simani, S. (2019). Novel non-model-based fault detection and isolation of satellite reaction wheels based on a mixed-learning fusion framework. *IFAC-PapersOnLine*. https://doi.org/10.1016/j.ifacol.2019.11.222

Probst, P., Wright, M. N., & Boulesteix, A. L. (2019). Hyperparameters and tuning strategies for random forest. *Wiley Interdisciplinary Reviews: Data Mining and Knowledge Discovery*, *9*(3), 1–15. https://doi.org/10.1002/widm.1301

Rahimi, A., Kumar, K. D., & Alighanbari, H. (2017). Fault estimation of satellite reaction wheels using covariance based adaptive unscented Kalman filter. *Acta Astronautica*, *134*, 159–169. https://doi.org/10.1016/j.actaastro.2017.02.003

Rahimi, A., & Saadat, A. (2019). Fault isolation of reaction wheels onboard 3-axis controlled in-orbit satellite using ensemble machine learning techniques. *The International Conference on Aerospace System Science and Engineering*.

Rahimi, A., & Saadat, A. (2020). Fault isolation of reaction wheels onboard three-axis controlled in-orbit satellite using ensemble machine learning. *Aerospace Systems*, *3*(2), 119–126. https://doi.org/10.1007/s42401-020-00046-x

Ramachandran, M., & Siddique, Z. (2019). A Data-Driven, Statistical Feature-Based, Neural Network Method for Rotary Seal Prognostics. *Journal of Nondestructive Evaluation, Diagnostics and Prognostics of Engineering Systems*. https://doi.org/10.1115/1.4043191

Sobhani-Tehrani, E., Talebi, H. A., & Khorasani, K. (2014). Hybrid fault diagnosis of nonlinear systems using neural parameter estimators. *Neural Networks*, *50*, 12–32. https://doi.org/10.1016/j.neunet.2013.10.005

Tidriri, K., Chatti, N., Verron, S., & Tiplica, T. (2016). Bridging data-driven and model-based approaches for process fault diagnosis and health monitoring: A review of researches and future challenges. *Annual Reviews in Control*, *42*, 63–81. https://doi.org/10.1016/j.arcontrol.2016.09.008

Biographies

Mofiyinoluwa Folami was born in Ontario, Canada in 1996. He received his B.Sc. degree in Mechanical Engineering at the University of Windsor in Canada in 2018. He began his M.Sc. in Mechanical Engineering at the University of Windsor in 2019 with an expected graduation date in 2021. During his B.Sc. degree he was involved in the 2018 Aerospace IREC Experimental Sound Rocket project., where he was responsible for the Aerostructure design and conducting Computation Fluid Dynamic simulations. For the last two years his research work was mainly focused on data-driven fault detection and diagnosis, machine learning and intelligent systems, with a specific study in the Attitude Determination and Control Subsystems of satellites systems.

Afshin Rahimi was born in Tehran, Iran in 1988. He received his B.Sc., M.Sc., and Ph.D. degrees in Aerospace Engineering from KNTU in Iran and Ryerson University in Canada in 2010, 2012, and 2017, respectively. He worked at Pratt & Whitney Canada from 2017 to 2018 and started an Assistant Professor position at the University of Windsor in 2018 in the Department of Mechanical, Automotive and Materials Engineering, where he resides now. For the last ten years, he has been involved in various industrial research, technology development, and systems engineering projects/contracts related to the control & diagnostics of satellites, UAVs, and commercial aircraft subsystems. His research work is primarily focused on model-based and data-driven fault detection, diagnostics, and prognosis; machine learning and intelligent systems; linear & nonlinear controller/observer design; and avionics, sensors, and measurement technologies. More recently, he has started working on smart factories and intelligent industrial automation in industry 4.0 in collaboration with industrial partners using computer vision, machine learning and artificial intelligence. He is a member of IEEE and AIAA institutions in addition to the PHM society.