

## Fault Classification for quadrotor UAV using spatial displacement statistical features.

Siti Noormiza binti Makhtar<sup>1,2</sup>, Umi Syahirah binti Mohd Sabudin<sup>1</sup>, Muhammad Harith bin Zaini<sup>1</sup>, Baizura binti Bohari<sup>3</sup>, Khairol Amali bin Ahmad<sup>1</sup>.

<sup>1</sup> Dept of Electrical and Electronic Engineering, Faculty of Engineering, National Defence University of Malaysia, Malaysia

<sup>2</sup> Dept of Industrial Revolution, Centre for Cyber Security & Digital Industrial Revolution, National Defence University of Malaysia, Malaysia

<sup>3</sup> Dept of Aeronautic Engineering and Aviation, Faculty of Engineering, National Defence University of Malaysia, Malaysia

---

### ARTICLE INFO

### ABSTRACT

#### *Article history:*

#### *Keywords:*

UAV; Machine Learning; Spatial displacement; Fault classification.

Spatial displacement statistical features are crucial parameters extracted from optical sensor data carrying informative imprints related to the UAV flying conditions. Incorporating these features into a flight faulty classification model for UAVs can provide insights into spatial variations to detect abnormalities and potential faults. This work integrates spatial displacement statistical analysis with a machine learning classification approach to propose a malfunction detection model for vertical take-off and landing quadrotor UAV. Spatial displacement data of the hovering quadrotor in healthy and faulty conditions are recorded using an optical camera system. Pertinent spatial parameters are retrieved using statistical analysis to provide informative features for the classification model. These spatial statistical features indicate hidden flying characteristics concealed by the raw signals captured during flight. The extracted features are used to train and classify different types of faults. The classifier model is also tested to validate its effectiveness in classifying faults from quadrotor UAVs with lower battery levels. The approach shows promising potential for real-time fault diagnosis using spatial displacement sensors to enhance the operational safety of the quadrotor UAV.

---

## 1. Introduction

Quadrotor unmanned aerial vehicles (UAVs) are compact aircraft that can fly slowly and land in confined spaces, which sets them apart from conventional aircraft. This type of UAV can be used for a variety of mission scenarios, including search and rescue, path planning [1], [2], performance evaluation[3], target tracking, and formation flight. Its multipurpose functionalities have attracted much attention in academic and commercial research in recent years. The reliability and safety of quadrotor UAV are prominent research topics within the scope of flight operations and predictive maintenance. Unexpected events, including actuator, sensor, or structural failures, can compromise the reliability and safety of vertical take-off and landing (VTOL) aircraft. Therefore, timely fault detection and identification (FDI) ensures flight safety [4]. Various diagnostic approaches have been explored, leveraging techniques such as machine learning [5], signal processing [6], and statistical signal analysis to detect and differentiate component failures [7].

VTOL aircraft refers to its ability to take off and land vertically without a runway. The VTOL UAV is a complex system with six degrees of freedom [8]. The system complexity refers to the capability of moving in three translational and three rotational degrees of freedom. The translational degrees of freedom include three movement pairs: forward & backward, left & right, and up & down. The three rotational degrees of freedom are pitch, roll, and yaw. These six degrees of freedom represent its ability to move and rotate freely in space using multiple rotors or propulsion systems.

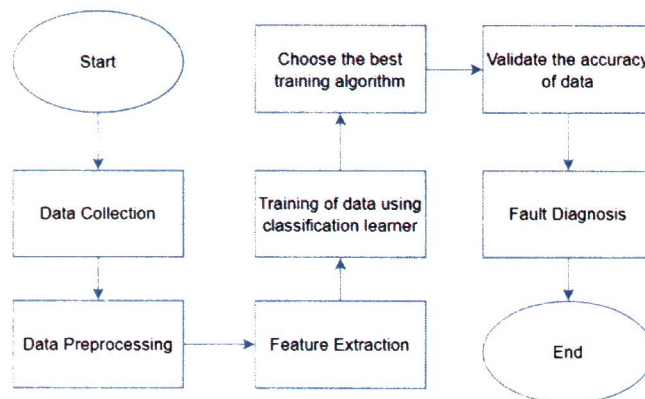
FDI is a multi-step process including fault detection, isolation, and estimation [9]. The first process of FDI aims to detect the occurrence of any fault using available flight data. Further investigation on multi-sensor data analysis will continue the process of fault isolation. The second process aims to identify UAV components that contribute towards flight failure. Finally, the fault estimation process will classify the severity of the specific component fault.

The application of data analysis from UAV sensor signals is beneficial for extracting informative features for diagnosing faults during flight operations. Extensive sensor data from the propulsion mechanism, flight control system and navigation signals provide beneficial information to identify patterns and anomalies during flight operation to facilitate FDI [10]. Integrating statistical signal processing techniques could enhance fault diagnosis accuracy by extracting valuable and informative features from extensive flight sensor data [11] in both time and frequency domains. The machine learning (ML) classification technique enables the development of predictive models to facilitate fast and early detection of flight faulty conditions potentially implemented for real-time FDI during flight operations.

The current project compares three ML classification algorithm to identify quadrotor propeller faulty conditions using statistical features of displacement data. Specifically, a statistical analysis of spatial positioning data for hovering quadrotor UAVs is analysed to identify informative features arising from propeller faults. The focal point lies in utilising machine learning algorithms to classify failures based on their distinctive translational motion characteristics.

## 2. Methodology

This section gives a clear and detailed description of the methods used to realise the objectives. Figure 1 shows a flow chart of this experimental study that begins with data collection, data preprocessing, feature extraction, training of classification learner, selection of the best classification algorithm and validation of the propeller faulty classification model.

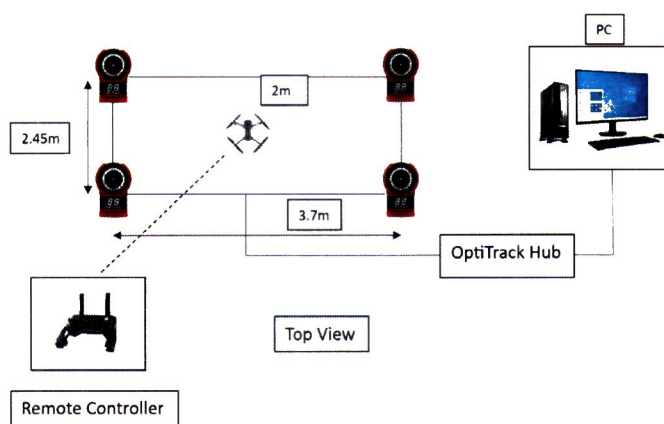


**Figure 1:** Flow chart of project methodology.

To diagnose faults in quadrotor UAVs, an indoor flight data collection process is designed to investigate significant anomalies from battery degradation and propeller failures. Data preprocessing is conducted to identify and extract relevant faulty features, ensuring data quality and suitability for subsequent analysis. This preprocessing phase encompasses data cleaning, handling missing values, and eliminating noise or outliers. Statistical signal processing techniques are then employed to perform feature extraction, aiming to identify and extract pertinent features associated with battery life and propeller fault. Subsequently, machine learning algorithms are leveraged to classify these extracted features and accurately identify faulty conditions. The fault diagnosis process is validated, and the performance of the classification algorithms is evaluated to assess their accuracy and facilitate in-depth analysis.

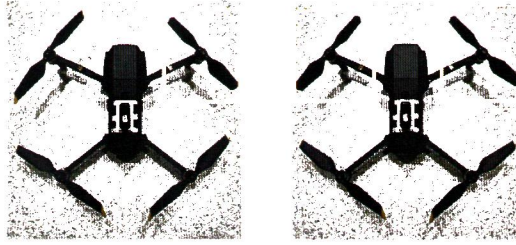
## 2.1 Data Collection

OptiTrack camera system captures the quadrotor's motion and transmits the flight movement data in x, y, and z coordinates using Motiv software. The OptiTrack camera system was installed and calibrated [12] in an indoor lab. As shown in Figure 2, four cameras were positioned in the lab to ensure a clear view of the reflective markers attached to the quadrotor. The system was configured to capture data at the desired frame rate and resolution[13]. Finally, experimental procedures are conducted to simulate different propeller faulty conditions. Experimental data recording emphasises the battery life [14] and broken propeller condition as faulty groups for further classification using the ML algorithm.



**Figure 2:** OptiTrack camera set-up for data collection.

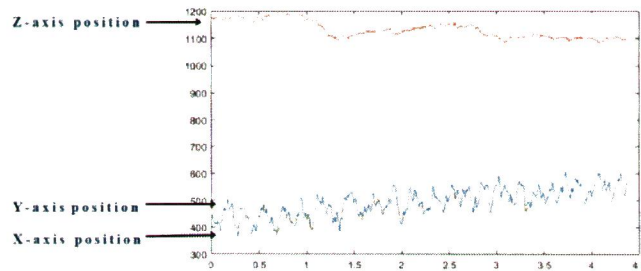
This study simulates two propeller faulty types to create six different classes of flying conditions with considerations on battery health. These classes include 1) standard propeller & high battery, 2) standard propeller & low battery, 3) one broken propeller & healthy battery, 4) one broken propeller & low battery, 5) two broken propellers & high battery, and 6) two broken propellers & low battery. Figure 3 illustrate the quadrotor with one and two broken propellers. This experimental setup is designed to provide valuable information about quadrotor UAVs' flying behaviour in different scenarios, contributing to a comprehensive classification for the FDI process. The flight data were imported into MATLAB workspace in position x, y, and z as shown in figure 3 below.



**Figure 3:** Quadrotor with one and two broken propellers

## 2.2 Data preprocessing

The recorded flight data were imported into the MATLAB workspace in positions  $x$ ,  $y$ , and  $z$ , as shown in Figure 4 below. The next essential step is to preprocess the recorded flight data and resampling to ensure the data is suitable for further statistical signal processing analysis. Data preprocessing is the application of filtering techniques to mitigate unwanted noise and interference [15], depending on the type of sensors used to collect the data. Data preprocessing [16] is crucial in preparing the collected data for subsequent feature extraction using statistical analysis. Preprocessing helps to enhance the data quality, extract meaningful features, and eventually enhance the accuracy of the classification model. ML training and validation using cleaner and more robust signal representation will produce better faulty classification algorithms using significant and informative features.



**Figure 4:** Recorded quadrotor tracking data in millimeters.

## 2.3 Feature extraction

Feature extraction as mentioned by Altinors *et al.*, [17] and Jae Hyeon *et al.*, [18] plays a crucial role in fault detection and analysis, capturing relevant information from collected data to uncover patterns and properties. Four key features [19] are commonly utilized in fault detection: kurtosis, mean, interquartile range (IQR) and standard deviation. Kurtosis is calculated for specific features or variables in a dataset. By considering the kurtosis of a feature, valuable insights can be gained regarding the concentration of values around the mean and the presence of outliers or extreme values. This information can help identify patterns or anomalies in the data. The IQR measures the spread of a dataset and is calculated as the difference between the data distribution's third quartile (Q3) and the first quartile (Q1). It quantifies performance variability and robustness in neural networks, providing insights into performance distribution across training runs or experimental conditions. Each vector element is divided by the RMS value for achieving normalization to detect any faulty as being said by Aleem *et al.* [20]. This is accomplished by dividing each element of the vector by the standard deviation. Standard deviation captures the dispersion of data points within a

feature and provides valuable information about the variability and consistency of the data. It is an important measure used in feature extraction for understanding the distribution and characteristics of the data. The formula in MATLAB for standard deviation is from equation 1, as the RMS formula is in equation 2 and the Interquartile range formula is in equation 3. These feature extraction techniques enable effective fault detection and analysis by capturing essential characteristics of the data in a standardized manner.

$$Std = \sqrt{\frac{\sum(x-\bar{x})}{n-1}} \quad (1)$$

$$K(x) = E \left[ \left( \frac{x-E(x)}{\sqrt{Var(x)}} \right)^4 \right] \quad (2)$$

$$IQR = Q3 - Q1 \quad (3)$$

#### 2.4 Quadrotor Faulty Classification

The Classification Learner toolbox in the Matlab software can run various classification algorithms to learn from the training data and analyse relationships between features and labels. The algorithms adjust parameters to optimise performance and minimise errors. Training data is a dataset consisting of input features and corresponding target labels. The data is structured in a tabular format, with each row representing an individual sample observation and each column representing a feature. Target labels indicate the class or category of each observation. During training, the model compares its predictions with known target labels, iteratively updating parameters. This process continues until the model achieves satisfactory accuracy or specified criteria. Once training is complete, the trained model can predict target labels for new instances based on their input features. Data quality and significant features are critical for the model's accuracy and generalisation [21]. Thus, careful data selection and preprocessing are essential before training in Classification Learner.

The training accuracy reflects the model's performance on the trained data, whereby the testing accuracy indicates the model's ability to generalise well to unseen instances and accurately classify them. The higher test accuracy suggests that the model has learned meaningful patterns from the training data and can effectively apply them to new instances. The following vital parameter to consider for building the classifier model is training time. This metric indicates the computational efficiency of the algorithm in training the model. A shorter training time implies faster model development, which can be advantageous when dealing with larger datasets or time-sensitive applications.

#### 2.5 Classification models performance.

The choice of the best training algorithm depends on various factors, including the characteristics of the dataset and the specific requirements of the problem at hand. Each algorithm has its strengths and weaknesses, and the best choice will depend on the specific context. To determine the best training algorithm, it is recommended to evaluate their performance on the specific dataset using appropriate metrics and cross-validation techniques [22]. Factors to consider include the size and complexity of the dataset, the desired interpretability of the model, the computational resources available, and any specific requirements or constraints of the problem. Evaluation from the scatter plot, confusion matrix, and ROC curve gives valuable insights into the model's accuracy, error

patterns, and discriminative power[23]. It plays an essential role in assessing the effectiveness of the trained model and guides further adjustments or improvements if necessary.

A scatter plot represents the predicted class labels compared to the actual class labels for a given set of test instances. Each point on the plot corresponds to an instance and shows the alignment between the predicted and actual class labels. Confusion matrix displays the number of true positive, true negative, false positive, and false negative predictions. The matrix arranges the actual class labels in rows and the predicted class labels in columns, facilitating detailed analysis of classification errors and accuracy. ROC curve plots the true positive rate against the false positive rate as the threshold varies. The ROC curve is useful for comparison between classes, with a higher area under the curve indicating superior performance.

### 3. Results

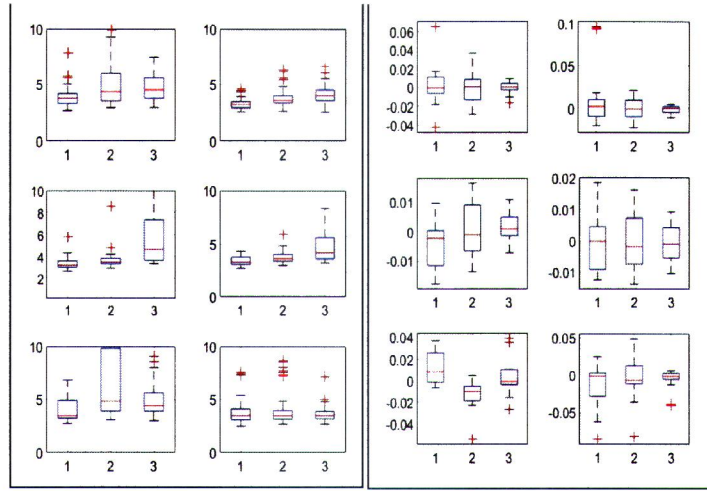
Recorded data is organised according to its battery and blade faulty condition. Each group will be visualised using linear plotting in MATLAB software for data preparation and cleaning. These initial findings provide valuable insights into the flying behaviour and response under different fault scenarios, contributing to a comprehensive understanding of its performance and durability. The flight data were imported into the MATLAB by column vectors and grouped into 6 groups as represented in Table 1.

**Table 1:** Organization of 6 groups of quadrotor flight data.

Group	Battery condition	Blade Faulty condition	Total samples
1	High	Normal	70
2	Low	Normal	72
3	High	Faulty 1	36
4	Low	Faulty 1	26
5	High	Faulty 2	36
6	Low	Faulty 2	80
Total			312

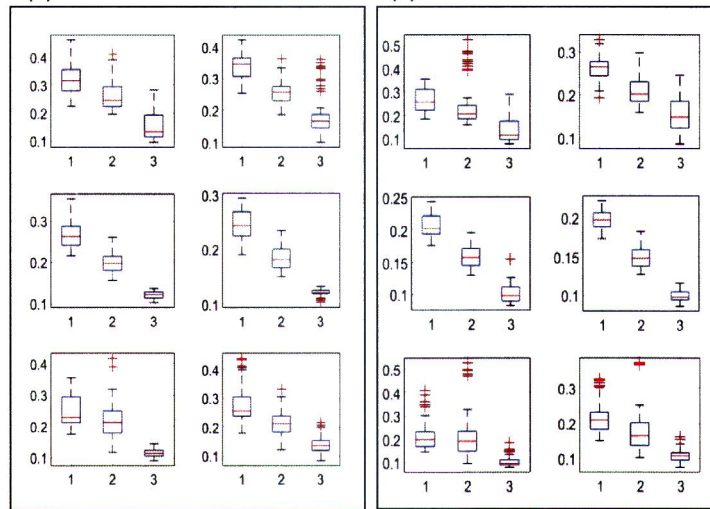
Four statistical features: kurtosis, mean, IQR and standard deviation were extracted from each of the flight tracking data axes: x, y, and z. To visualize the distribution of these features across six different groups, boxplots were generated as shown in Figure 5. These boxplots provide a visual representation of the distribution of the extracted features across the six different groups, allowing for a comparative analysis and insights into the variations and patterns within the data sets. The labels 1, 2, and 3 in these figures represent the different distributions observed for the x, y, and z datasets, respectively.

The ML model evaluations are employed for assessing the performance of three classification algorithms: Quadratic Support Vector Machine (Q-SVM), Kernel Naïve Bayes (K-NB) and Fine Tree (F-T) algorithms. A scatter plot in Figure 6 shows comparison between the predicted class labels with the actual class labels for randomly selected pair of statistical features in Q-SVM classification model. This scatter plot visually represents the alignment between the predicted and actual class labels, enabling the assessment of the model's accuracy by examining the degree of agreement between the predicted and true labels.



(a) kurtosis values

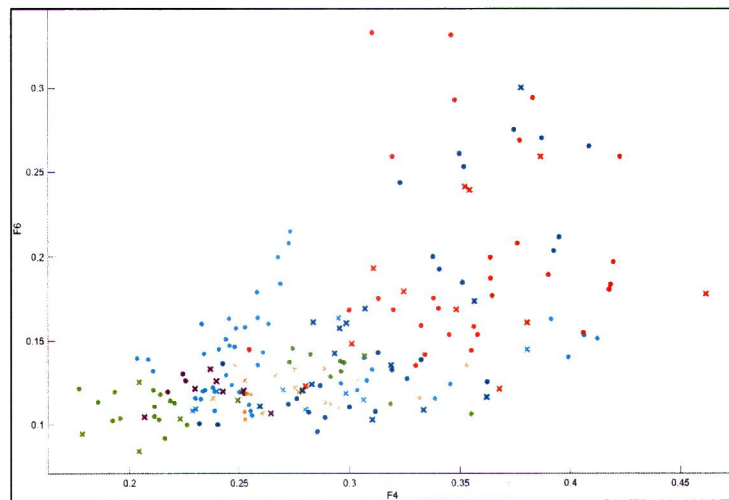
(b) mean values



(c) interquartile range values

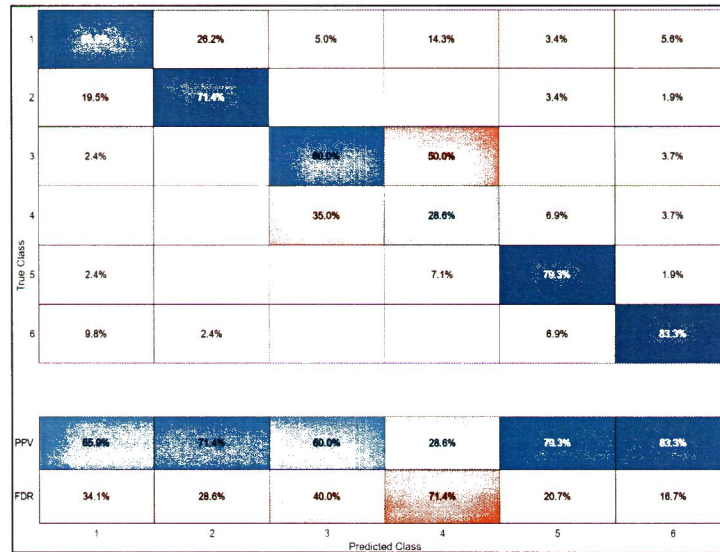
(d) standard deviation

**Figure 5:** Boxplot of statistical features from six different group of data



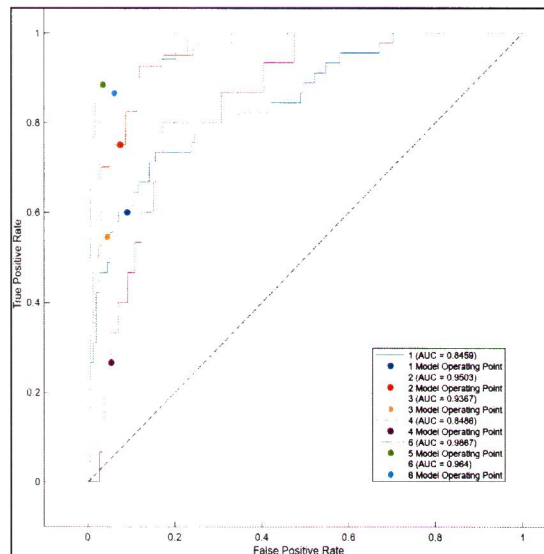
**Figure 6:** A scatter plot of predicted class labels compared with the actual class labels.

The confusion matrix in Figure 7 offers a comprehensive overview of the Q-SVM model's performance. It presents the percentage of true (blue box) and false (brown box) predictions with color spectrum range for a better visual observation on the accuracy levels.



**Figure 7:** Confusion matrix of true predictions (blue box) and false predictions (brown box) accuracy percentage.

The ROC curve in Figure 8 is employed to illustrate the performance of the Q-SVM model across various classification thresholds. This curve plots the true positive rate (sensitivity) against the false positive rate (1 - specificity) as the threshold varies. The area under the ROC curve (AUC) is then used to determine the discriminative power of the model, with a higher AUC indicating superior performance in class discrimination.



**Figure 8:** ROC curve to illustrate the performance of the model across classification thresholds

Table 2 presents the results from the Classification Learner for Q-SVM, K-NB, and F-T algorithms. For the first model, Q-SVM can correctly classify 70.5% of instances within the training dataset. On the test data, the Q-SVM model achieved a slightly reduced accuracy of 66.1%, indicating its good generalisation performance and ability to classify unseen instances. The training time for Q-SVM is 2.139 seconds, showcasing its computational efficiency in building the classification model. In the next model, K-NB classification accuracy of the train data is slightly lower than Q-SVM, with 66% correct classification. On the test data, the K-NB model exhibited decent generalisation performance with the accuracy of 61.6%, indicating its medium level of effectiveness in accurately classifying new instances. However, the training time for K-NB is significantly longer than the previous model. It takes 4.6239 seconds to construct the classification model. Finally, the F-T algorithm achieved the lowest accuracy of 52.0% on the train data. It demonstrated less reliable generalisation performance with a testing accuracy of 50.9% and a long training time of 6.0137 seconds.

**Table 2:** Accuracy levels for three selected classification algorithms

Classification Model Accuracy (%)		
Algorithms	Train	Test
Quadratic SVM	70.5	66.1
Kernel NB	66.0	61.6
Fine Tree	52.0	50.9

The crucial phase of statistical signal processing has proven its ability to capture underlying fault patterns and detect possible failures caused by faulty blades and degraded battery performance. Initial data preprocessing techniques, including visual observations of the datasets, are also essential in eliminating artefacts from raw data, producing significant features for each data group. The selection and extraction of relevant features using statistical signal processing techniques provide valuable information for faulty group classification.

While the fault diagnosis methodology using statistical signal processing for quadrotor UAVs offers promising results, it is essential to acknowledge its limitations and consider potential improvement possibilities. One limitation is the reliance on displacement data for fault detection and diagnosis. The accuracy and effectiveness of the methodology heavily depend on the quality and reliability of the recorded optical image data. Any measurement limitations or errors will impact the overall performance of the faulty classification. Another limitation is the fault classification algorithm's dependency on labelled training data. Insufficient or imbalanced training data may result in reduced accuracy and limited generalisation ability of the classification algorithm.

For further improvements, expanding and refining the labelled training dataset can increase the fault classification algorithm's performance by conducting more comprehensive experiments and incorporating a wider range of fault scenarios. Additionally, considering complementary analysis methods, such as deep learning approaches, can enhance fault diagnosis capabilities, especially for complex and subtle faults. These techniques can capture intricate fault patterns and relationships that may not be easily discernible through traditional statistical signal processing methods.

#### 4. Conclusions

In conclusion, Q-SVM shows the best classification performance and is considered an outstanding model for developing a reliable fault diagnosis system for the real- world implementation of faulty UAV prediction systems using spatial displacement data. Statistical signal processing has proven its ability to capture informative features from different propeller failures and degraded battery

performance. Several suggestions should be considered to develop a real-time fault diagnosis system. Reliable and robust vibration sensors can improve classification performance in comprehensive experiments incorporating a wider range of fault scenarios.

### Acknowledgement

This project is supported by Fundamental Research Grant Scheme: FRGS/1/2022/TK07/UPNM/02/10 from The Malaysia Ministry of Higher Education.

### References

- [1] Galvez, Reagan L., Elmer P. Dadios, and Argel A. Bandala. "Path planning for quadrotor UAV using genetic algorithm." In *2014 International Conference on Humanoid, Nanotechnology, Information Technology, Communication and Control, Environment and Management (HNICEM)*, pp. 1-6. IEEE, 2014. <https://doi.org/10.1109/HNICEM.2014.7016260>
- [2] Allaire, François CJ, Mohamed Tarbouchi, Gilles Labonté, and Giovanni Fusina. "FPGA implementation of genetic algorithm for UAV real-time path planning." In *Unmanned Aircraft Systems: International Symposium On Unmanned Aerial Vehicles, UAV'08*, pp. 495-510. Springer Netherlands, 2009. [https://doi.org/10.1007/978-1-4020-9137-7\\_26](https://doi.org/10.1007/978-1-4020-9137-7_26)
- [3] Abeywickrama, Hasini Viranga, Beeshanga Abewardana Jayawickrama, Ying He, and Eryk Dutkiewicz. "Comprehensive energy consumption model for unmanned aerial vehicles, based on empirical studies of battery performance." *IEEE access* 6 (2018): 58383-58394. <https://doi.org/10.1109/ACCESS.2018.2875040>
- [4] Zhang, Xiaomin, Zhiyao Zhao, Zhaoyang Wang, and Xiaoyi Wang. "Fault detection and identification method for quadcopter based on airframe vibration signals." *Sensors* 21, no. 2 (2021): 581. <https://doi.org/10.3390/s21020581>
- [5] Maxwell, Aaron E., Timothy A. Warner, and Fang Fang. "Implementation of machine-learning classification in remote sensing: An applied review." *International journal of remote sensing* 39, no. 9 (2018): 2784-2817. <https://doi.org/10.1080/01431161.2018.1433343>
- [6] Delgado-Arredondo, Paulo Antonio, Daniel Morinigo-Sotelo, Roque Alfredo Osornio-Rios, Juan Gabriel Avina-Cervantes, Horacio Rostro-Gonzalez, and Rene de Jesus Romero-Troncoso. "Methodology for fault detection in induction motors via sound and vibration signals." *Mechanical Systems and Signal Processing* 83 (2017): 568-589. <https://doi.org/10.1016/j.ymssp.2016.06.032>
- [7] Dutta, Airin, Michael McKay, Fotis Kopsaftopoulos, and Farhan Gandhi. "Statistical residual-based time series methods for multicopter fault detection and identification." *Aerospace Science and Technology* 112 (2021): 106649. <https://doi.org/10.1016/j.ast.2021.106649>
- [8] V. Gomez, N. Gomez, J. Rodas, E. Paiva, M. Saad, and R. Gregor, "Pareto optimal PID tuning for Px4-based unmanned aerial vehicles by using a multi-objective particle swarm optimization algorithm," *Aerospace*, vol. 7, no. 6, 2020, doi: 10.3390/AEROSPACE7060071.
- [9] A. Fekih, "Fault diagnosis and Fault Tolerant Control design for aerospace systems: A bibliographical review," 2014 American Control Conference, Portland, OR, USA, 2014, pp. 1286-1291, doi: 10.1109/ACC.2014.6859271.
- [10] G. Ortiz-Torres, P. Castillo, and J. Reyes-Reyes, "An Actuator Fault Tolerant Control for VTOL vehicles using Fault Estimation Observers: Practical validation," 2018 Int. Conf. Unmanned Aircr. Syst. ICUAS 2018, pp. 1054-1062, 2018, doi: 10.1109/ICUAS.2018.8453288.
- [11] A. Altinors, F. Yol, and O. Yaman, "A sound based method for fault detection with statistical feature extraction in UAV motors," *Applied Acoustics*, vol. 183, 2021, doi: 10.1016/j.apacoust.2021.108325.
- [12] Gomez, Victor, Nicolas Gomez, Jorge Rodas, Enrique Paiva, Maarouf Saad, and Raul Gregor. "Pareto optimal pid tuning for Px4-Based unmanned aerial vehicles by using a multi-objective particle swarm optimization algorithm." *Aerospace* 7, no. 6 (2020): 71. <https://doi.org/10.3390/aerospace7060071>
- [13] Chen, Qing, Yu Zhou, Yong Wang, Mao Mao Zhu, Lei Guo, and Chang Xu He. "Research on stability and accuracy of the OptiTrack system based on mean error." In *International Symposium on Artificial Intelligence and Robotics 2021*, vol. 11884, pp. 438-450. SPIE, 2021. <https://doi.org/10.1117/12.2605796>
- [14] Bello, Ana Belén, Francisco Navarro, Javier Raposo, Mónica Miranda, Arturo Zazo, and Marina Álvarez. "Fixed-Wing UAV Flight Operation under Harsh Weather Conditions: A Case Study in Livingston Island Glaciers, Antarctica." *Drones* 6, no. 12 (2022): 384. <https://doi.org/10.3390/drones6120384>
- [15] Yu, Jianbo, Tianzhong Hu, and Haiqiang Liu. "A new morphological filter for fault feature extraction of vibration signals." *IEEE Access* 7 (2019): 53743-53753. <https://doi.org/10.1109/ACCESS.2019.2912898>

- [16] A. Bondyra, P. Gasiór, S. Gardecki, and A. Kasinski, "Fault diagnosis and condition monitoring of UAV rotor using signal processing," *Signal Process. - Algorithms, Archit. Arrange. Appl. Conf. Proceedings, SPA*, vol. 2017-Septe, pp. 233–238, 2017, doi: 10.23919/SPA.2017.8166870.
- [17] Altınors, Ayhan, Ferhat Yol, and Orhan Yaman. "A sound based method for fault detection with statistical feature extraction in UAV motors." *Applied Acoustics* 183 (2021): 108325. <https://doi.org/10.1016/j.apacoust.2021.108325>
- [18] Park, Jae-Hyeon, Cho-Yong Jun, Jin-Yeong Jeong, and Dong Eui Chang. "Real-time quadrotor actuator fault detection and isolation using multivariate statistical analysis techniques with sensor measurements." In *2020 20th International Conference on Control, Automation and Systems (ICCAS)*, pp. 33-37. IEEE, 2020. <https://doi.org/10.23919/ICCAS50221.2020.9268391>
- [19] Benini, Alessandro, Francesco Ferracuti, Andrea Moneriù, and Stephan Radensleben. "Fault detection of a VTOL UAV using acceleration measurements." In *2019 18th European Control Conference (ECC)*, pp. 3990-3995. IEEE, 2019. <https://doi.org/10.23919/ECC.2019.8796198>
- [20] Aleem, Saad Abdul, Nauman Shahid, and Ijaz Haider Naqvi. "Methodologies in power systems fault detection and diagnosis." *Energy Systems* 6 (2015): 85-108. <https://doi.org/10.1007/s12667-014-0129-1>
- [21] Oquadine, Ahmed Youssef, Mostafa Mjahed, Hassan Ayad, and Abdeljalil El Kari. "UAV quadrotor fault detection and isolation using artificial neural network and Hammerstein-Wiener model." *Studies in Informatics and Control* 29, no. 3 (2020): 317-328. <https://doi.org/10.24846/v29i3y202005>
- [22] Wang, Xiangwen, Xianghong Lin, and Xiaochao Dang. "Supervised learning in spiking neural networks: A review of algorithms and evaluations." *Neural Networks* 125 (2020): 258-280. <https://doi.org/10.1016/j.neunet.2020.02.011>
- [23] Janiesch, Christian, Patrick Zschech, and Kai Heinrich. "Machine learning and deep learning." *Electronic Markets* 31, no. 3 (2021): 685-695. <https://doi.org/10.1007/s12525-021-00475-2>