# Deep-Learning fault detection and classification on a UAV propulsion system

Pierre-Yves BRULIN[1,2], Fouad KHENFRI[1], Nassim RIZOUG[1]
[1]ESTACA'Lab, S2ET
Parc universitaire Laval-Changé, Rue Georges Charpak, 53061 Laval, France
Email: first-name.last-name@estaca.fr
URL: http://www.estaca.fr
[2]HEXADRONE
ZA La Sagne, 99 Chem. de la Borie, 43330 Saint-Ferréol-d'Auroure, France
Email: pierre-yves.brulin@hexadrone.fr
URL: http://www.hexadrone.fr

## Keywords

≪Fast Fault Detection≫, ≪Deep Learning≫, ≪Machine Learning≫, ≪Permanent magnet motor≫, ≪Condition Monitoring≫

## Abstract

A fault detection and identification method using a Deep-Learning classification method is used to identify several faults that may occur on a UAV propulsion system. Training is performed from a dataset acquired from a simplified multiphysics simulation of the system which allows for the generation of large datasets of modular, interconnected and scalable components of various sizes and performances. We aim to provide a model able to identify faults occurring on a propulsion system using a reduced set of input signals.

## Introduction

Drones or swarm of drones are now gaining interest in several industrial applications such as package delivery, environmental monitoring, aerial photography, 3D mapping, industrial and infrastructure inspections, and much more. The growth in the use of drones involves many complex independent systems to monitor. On a large fleet, human interventions on each machine are necessary to maintain its performances and integrity. On the other hand, every levels of maintenance cannot be performed by all the operators of the same machine. Fig. 1 shows the five levels of maintenance as described in [1]. The 1st and 2nd maintenance level performed respectively by the end-user and by an authorized technician remains limited in the actions that can be performed on the machine and does not involve any action on its internal components. Beyond the 3rd maintenance level, the diagnosis of failures is considered but implies specialized knowledge of the system and allow a repair limited to the exchange of components. Beyond the 4th level, the system is entrusted to a dedicated intervention team, or sent back to the manufacturer, for a partial or total reconstruction.

An internal system fault can only be identified after an examination by a specialized operator at the 3rd level. At this level, the cost and time required for maintenance are very high and can be subject to errors in the identification of defective components. In order to overcome this problem, different methods are developed to detect and isolate faults (FDI) that occur on complex and critical systems (i.e. airplanes, cars, drones, ...). In [2], fault detection (FD) and fault isolation (FI) methods are classified into 3 categories: hardware redundancy, signal processing and analytical redundancy.

The hardware redundancy method is not applicable to the propulsion system due to the limitations of UAV in terms of their carrying capacity which drastically limits the number of subsystems that can be

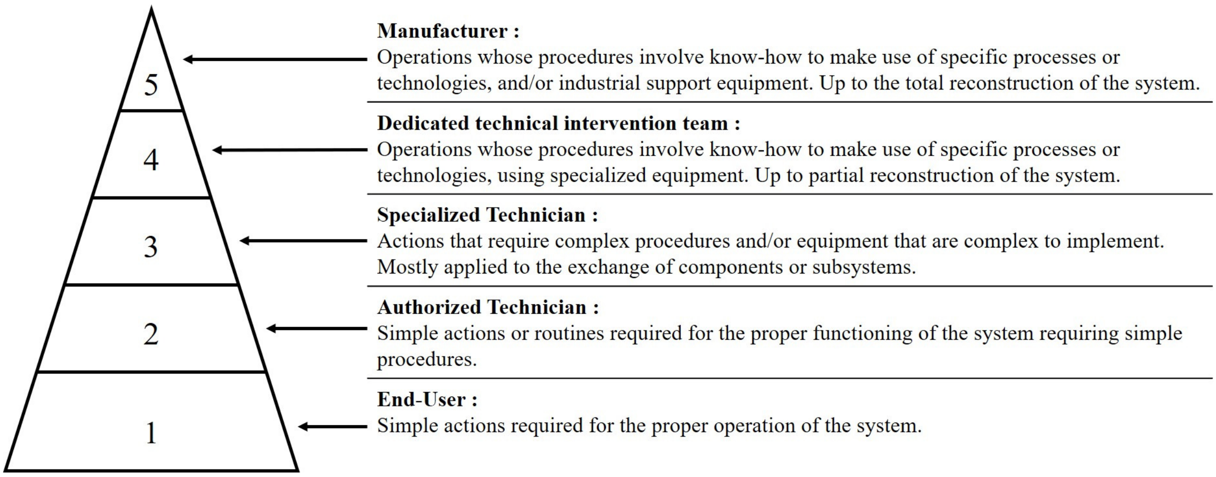


Fig. 1: Hierarchy of the five levels of maintenance according to [1]

installed on the same machine. Thus this method is mainly limited to the sensors that are critical to the attitude control of the machine for which it demonstrates a strong capacity to isolate faults on these components.

The signal processing method and the model-based analytical redundancy method are numerous as shown in [3], but are mainly restricted to a predefined system and limit the scalability of the considered FDI system. [4] and [5] propose the detection of faults on a PMSM by the real-time comparison between a simulated model and the measurements of a motor on a test bench. The presented procedure consists in the prior identification of the propulsion system parameters in order to obtain a healthy representation close to reality. The detection of faults is then carried out by comparing the measured and simulated signals to detect any anomaly on the system operation. For [5], the study focuses on the speed and the current flowing through the motor input. In [4] the measured currents from the battery and the stator phases of the motor, as well as their frequency in operation are studied. This frequency study of the currents is similar to [6] which studies the stator phase currents for the monitoring of open phase faults, using only current sensors for the detection. This work relies mainly on the use of flowcharts designed to identify a behavior that would not follow the expectation of a healthy system. Finally, [7] proposes the determination of faults related to the eccentricity of the motor and the propeller, based on the frequency analysis of the system. The signal is then compared to trend curves to determine if the system is balanced or not.

In this paper, we rely on the data-driven analytical redundancy method to maximize the fault detection and classification processing so that our solution is not restricted to a single specific system configuration. Several recent and relevant research works focus on the data-driven analytical method [8-11]. In [8], the authors use fuzzy logic to detect the state of the motor and its propeller by combining a frequency study of the stator phase currents for faults that may occur on the motor, and the accelerations at the end of the motor block for faults on the propeller. Authors of [9] proposes a method for identifying faults that may occur on the propellers of a multirotor UAV based on the frequency study of vibrations on the UAV chassis using the accelerometer sensors already used by the flight controller. In [10] an embedded deep-learning processing is used to detect faults that could occur to the whole UAV system and whose natures are very varied, since the behaviors to be identified also include the risks of cyber-attacks. The presented two-step logic works initially by detecting if there is a faulty behavior ongoing by using a CNN-BiLSTM model in an encoder-decoder logic for unsupervised learning, and then if a fault is detected to classify it. This processing uses several temporal data from the UAV's onboard sensors (IMU, accelerometers, gyroscope, etc.) to identify specific patterns and temporal dependencies that would demonstrate a fault impacting the UAV. Finally [11] presents the CNN-BiLSTM architecture model as a preferred detection method for the analysis of temporal signals from complex industrial machines.

We seek to ensure reliable detection and identification of faults that may occur on propulsion system by minimizing the sensors required to operate our fault detection in the face of cost and weight constraints on the UAV.

This paper is organized as follows: we will first present the generation of our dataset from a simplified multiphysics model to collect the faulty behaviors to be detected. Then, we will present the neural network architecture able to answer the problem.

## Generation of the training dataset

We focus on the study of a Permanent Magnet Synchronous Motor (PMSM) equipped with a propeller for propulsion and its Electronic Speed Controller (ESC) driver, allowing us not to restrict this application to multirotor UAVs only. Nonetheless the case of multirotor UAVs allows a wide application setting, since we will find several instances of this independent system according to the multirotor configuration. The use of this type of motor equipped with a propeller can also be shared with other propulsion systems as the ones of fixed-wing UAVs, or even surface vehicles (USV) such as hovercrafts.

As fault data is scarce on this type of system and the cost of each components to allow the acquisition of this type of data is high, the use of real components to collect fault states, which will result in the destruction of the component, is not viable. In this perspective, we propose the use of a simplified simulated model of this propulsion system in which we can virtually inject faults of different natures and generate as many scenarios as necessary.

The model of the propulsion system is composed of the following components:

1. The battery for powering the components;
2. The Electronic Speed Controller (ESC) which controls the motor;
3. The Permanent Magnet Synchronous Motor (PMSM);
4. The propeller allowing the conversion of the rotation speed into propulsion force.

All these components are integrated into a multiphysics model using simplifying assumptions to simulate both the electrical and thermal behavior of each components. We use the observed performance of the system to generate a dataset of temporal signals for the fault-induced responses.

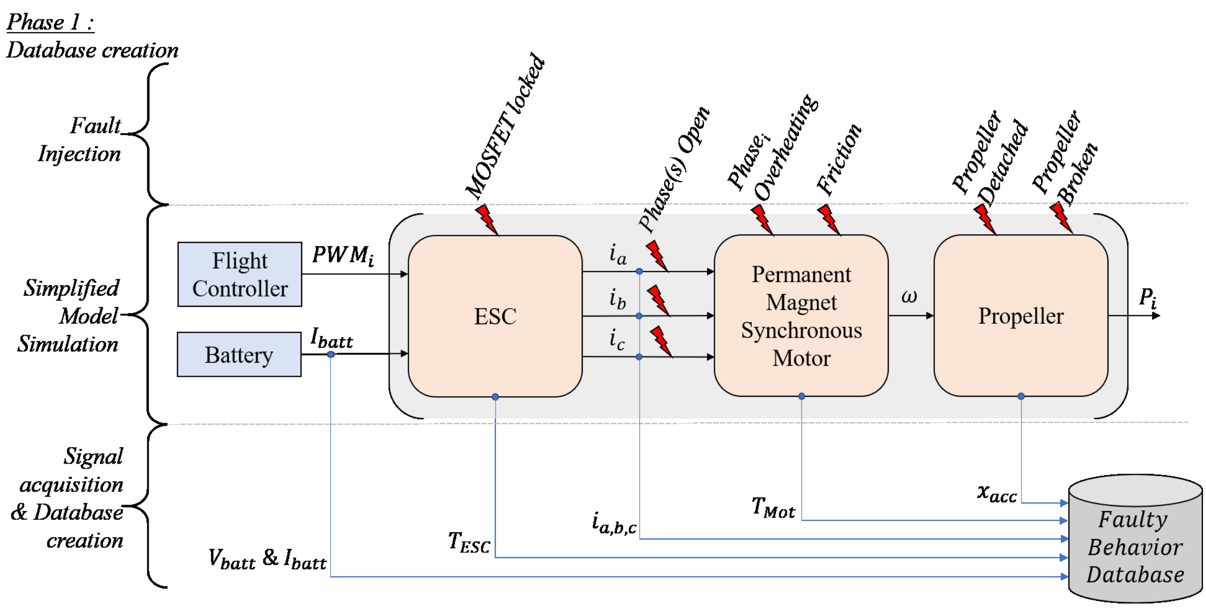


Fig. 2: Creation of the dataset of faulty behaviors from the simulated propulsion system

Fig. 2 shows the initial generation of our dataset of fault-prone behaviors. The authors of [4] and [5] present a list of faults that can occur on a propulsion system. We choose to inject the following faults one at a time in our model to create our dataset:

- On the Electronic Speed Controller:

- One of the MOSFETs cannot commutate;
- On the Permanent Magnet Synchronous Motor:
  - One of the stator phases is open;
  - Excessive overheating of one of the motor phases (causing the internal resistance of the phase to increase);
  - The rotor is subject to abnormal friction (e.g. damaged bearings or friction);
- On the propeller:
  - The propeller is detached (the motor is running idle);
  - The propeller has a break, inducing an eccentricity on the rotor.

On our simplified model, we simulate the initial causes of each fault considered and we observe all the signals selected on the system following the introduction of this fault. This allows us to make correlations between the injected faults and the system behavior. We also make the initial assumption that only one fault can occur at a time in our model.

In order to perform the detection and identification of each of these faults, we acquire and store in our dataset the following signals at different input throttle commands:

- The voltage and current drawn at the ESC input;
- The currents of each phase of the motor;
- The accelerations perceived at the rotor;
- The motor temperature;
- The temperature of the ESC, whose power dissipation comes mainly from the MOSFETs that allow the current flow in each phase of the stator (conduction losses due to their resistance, losses induced by the switching associated with the control frequency of the FET and a set of losses resulting from the recirculation of currents during switching [12]);

Due to the nature of the components and sensors assumed for signal collection, we also introduce signals that vary at random frequencies and amplitudes, in order to simulate spurious and background noise that will be superimposed on the signal.

We present in Table I the characteristics of one of the propulsion systems whose parameters we apply to our simulated model for the creation of our dataset.

Table I: Example of a combination of characteristics of components used in the propulsion system

| Component | Characteristic | Value |
|---|---|---|
| ESC: T-Motor Alpha 60A | Input Voltage | 25.2 V |
| | Peak Current | 60 A |
| Motor: T-Motor U8II 190KV | Configuration | 36N42P |
| | Rs | $48 \pm 3m\Omega$ |
| | $\Psi_m$ | $1.484e^{-3}Wb$ |
| | Ipeak | 43.7 A |
| | Ppeak | 1048.8 W |
| Propeller: T-Motor MF2815 | Diameter | 28.4” / $721.8e^{-3}m$ |
| | Max Thrust / RPM | 15 kg / 5000 RPM |
| | $C_T$ / $C_Q$ | 2e-5 / 75e-8 |

## Training and Fault Detection

Fig. 3 shows the neural network training and detection phase applied on a real system components whose characteristics were used as parameters in the simulation for the acquisition. The training is done entirely from the timeseries recorded in the dataset. For training our neural network, we reserve one third of our acquired dataset for validation while the rest will be used to train the model.

The input of our model consists of windows of 250 elements per the number of signals. Since faults are manually injected into our simulated model, we can also use the transient phases of our fault-prone

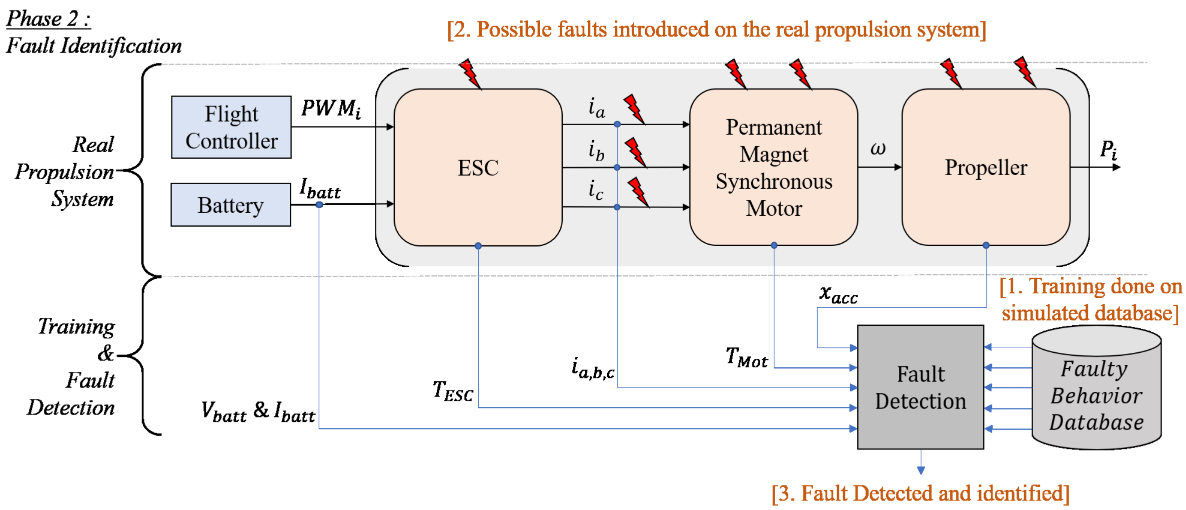


Fig. 3: Training of the fault detection system via the recorded dataset and identification of the faults occurring on a real system

system for training. In order to label each window for the classification training, we consider that a window of signals is showing a fault if one has been injected during more than 75% of the window acquisition.

Since the signals measured from our model are of different natures, we must adapt the data in order to standardize the inputs of our model. We normalize our data by window and not on the whole dataset. This is necessary, because some faults may cause some of the captured signals to diverge. Since our model aims to fit different components configurations, the data cannot be normalized between specific minimum/maximum values for every configurations. Thus, we normalize our sample data between the maximum values of each signal window.

$$X^*_{i\text{window}} = \frac{X_{i\text{window}}}{\max\left(\left|X_{i\text{window}}\right|\right)} \tag{1}$$

In healthy operation, we expect to observe similar or repeatable patterns on the signals, while defective behavior should cause deviations identifiable by the layers of our neural network.

## Neural Network Architecture

Fault detection relies on the use of a deep-learning multiheaded model based on one-dimensional CNN (Convolutional Neural Network) and LSTM (Long-Short Term Memory) layers as shown in Fig. 4.

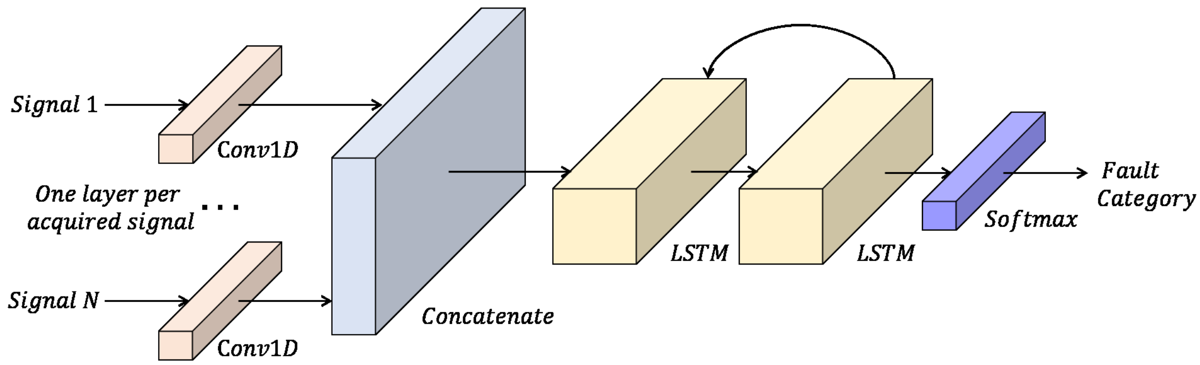


Fig. 4: Neural network architecture for fault detection and identification

The structure of our neural network based on a multiheaded model allows our model to receive input data of diverse nature from different sensors, where each head is able to extract the features of each signal independently. The input of our model can be considered as multivariate timeseries, therefore one-dimensional CNN layers are used as input layers at each one of the model heads. A one-dimensional Convolutional layer operates by drawing smaller sequences of data from long one-dimensional sequences ready to be interpreted by downstream layers.

After concatenation, the final Bidirectionnal-LSTM layers can determine the long-term dependencies of the model. Recurrent Neural Networks (RNN) are subject to the long-term problem of gradient dissipation during model training. This means that traditional RNNs cannot natively capture long-term dependencies for weight update. LSTMs were designed as to preserve both short and long-term network memories, to prevent backpropagation errors from disappearing or exploding. Capable of capturing dynamic timeseries states of systems, LSTMs have been successfully applied in various applications such as speech recognition, handwriting recognition and natural language processing.

The training is performed using the Adam optimizer and the categorical crossentropy as the loss function. Input windows are sequentially fed into our neural network using the sliding window approach, which is based upon using the previous system states to reinforce the temporal characteristics and dependencies of our input signals caught by the underlying layers.

## Performance evaluation

The classification allows us to identify faults that may occur in the propulsion system. In Fig. 5, we present the confusion matrix reflecting the challenges our classifier faces in differentiating between healthy and faulty system states. Our neural network performs well on a balanced validation dataset which contains more than 1500 sample windows for each state.

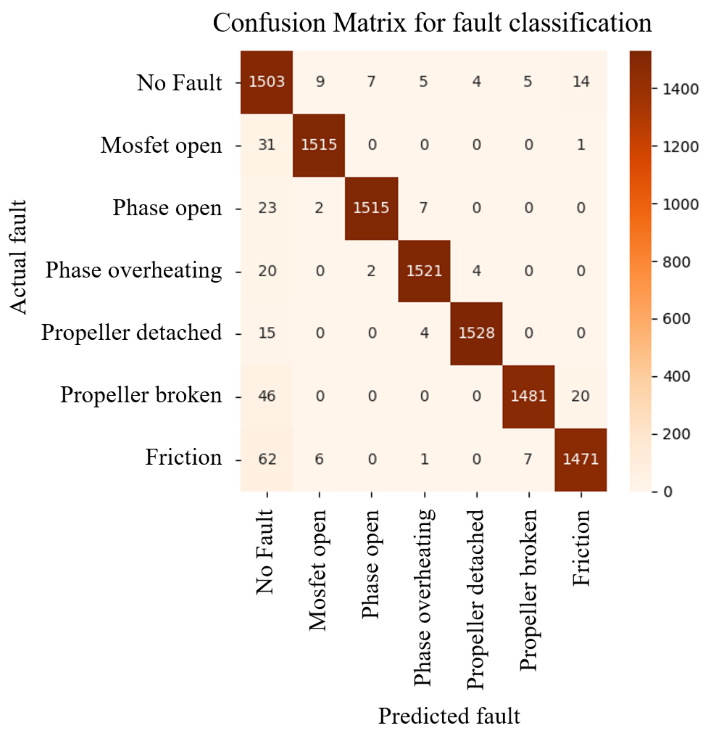


Fig. 5: Confusion matrix for the fault classifier on the validation dataset

We observe that the case where the model performs the worst corresponds to the identification of the friction fault on the motor as opposed to the healthy state. This is due to the fact that we have introduced different levels of possible friction on our simulated model, sometimes at low levels that are difficult for our neural network classifier to perceive. Furthermore, the relative confusion between the healthy and faulty states by our classifier is mainly explained by the addition of the transient phases of the faulty system for the training.

# Conclusion

We present a method for detecting and identifying faults in the components of a UAV propulsion system. The data acquired from a multiphysics model using simplifying assumptions allows us to model the electrical and thermal behavior of the system when it is affected by a fault. This dataset generation method allows us to generate a balanced set of samples for training a deep learning neural network classifier to classify the different degraded behaviors of the system.

The multiheaded CNN-BiLSTM architecture allows us to perform a robust classification of the faulty behaviors from a reduced set of signals and sensors on the system, minimizing the cost and the load constraint on an actual machine.

We plan to apply this method on real components on a test bench in order to validate the detection of real faults introduced on the propulsion system.

# References


[1] "NF X60-000," *Afnor EDITIONS*. https://www.boutique.afnor.org/fr-fr/norme/nf-x60000/maintenance-industrielle-fonction-maintenance/fa063074/1561 (accessed Mar. 21, 2022).

[2] G. K. Fourlas and G. C. Karras, "A Survey on Fault Diagnosis and Fault-Tolerant Control Methods for Unmanned Aerial Vehicles," *Machines*, vol. 9, no. 9, p. 197, Sep. 2021, doi: 10.3390/machines9090197.

[3] S. Nandi, H. A. Toliyat, and X. Li, "Condition Monitoring and Fault Diagnosis of Electrical Motors—A Review," *IEEE Trans. Energy Convers.*, vol. 20, no. 4, pp. 719–729, Dec. 2005, doi: 10.1109/TEC.2005.847955.

[4] G. Jouhet, L. E. González-Jiménez, M. A. Meza-Aguilar, W. A. Mayorga-Macías, and L. F. Luque-Vega, "Model-Based Fault Detection of Permanent Magnet Synchronous Motors of Drones Using Current Sensors," in *New Trends in Robot Control*, J. Ghommam, N. Derbel, and Q. Zhu, Eds. Singapore: Springer, 2020, pp. 301–318. doi: 10.1007/978-981-15-1819-5_15.

[5] J. Lee, W. Lee, S. Ko, and H. Oh, "Fault Classification and Diagnosis of UAV motor Based on Estimated Nonlinear Parameter of Steady-State Model," *IJMERR*, pp. 22–31, 2020, doi: 10.18178/ijmerr.10.1.22-31.

[6] A. Khlaief, M. Boussak, and M. Gossa, "Open phase faults detection in PMSM drives based on current signature analysis," in *The XIX International Conference on Electrical Machines - ICEM 2010*, Rome, Italy, Sep. 2010, pp. 1–6. doi: 10.1109/ICELMACH.2010.5607977.

[7] F. C. Veras, T. L. V. Lima, J. S. Souza, J. G. G. S. Ramos, A. C. Lima Filho, and A. V. Brito, "Eccentricity Failure Detection of Brushless DC Motors From Sound Signals Based on Density of Maxima," *IEEE Access Pract. Innov. Open Solut.*, vol. 7, pp. 150318–150326, 2019, doi: 10.1109/ACCESS.2019.2946502.

[8] F. Pourpanah, B. Zhang, R. Ma, and Q. Hao, "Anomaly Detection and Condition Monitoring of UAV Motors and Propellers," in *2018 IEEE SENSORS*, Oct. 2018, pp. 1–4. doi: 10.1109/ICSENS.2018.8589572.

[9] X. Zhang, Z. Zhao, Z. Wang, and X. Wang, "Fault Detection and Identification Method for Quadcopter Based on Airframe Vibration Signals," *Sensors*, vol. 21, no. 2, p. 581, Jan. 2021, doi: 10.3390/s21020581.

[10] V. Sadhu, S. Zonouz, and D. Pompili, "On-board Deep-Learning-Based Unmanned Aerial Vehicle Fault Cause Detection and Identification," May 06, 2020. http://arxiv.org/abs/2005.00336 (accessed Apr. 12, 2021).

[11] M. Canizo, I. Triguero, A. Conde, and E. Onieva, "Multi-head CNN–RNN for multi-time series anomaly detection: An industrial case study," *Neurocomputing*, vol. 363, pp. 246–260, Oct. 2019, doi: 10.1016/j.neucom.2019.07.034.

[12] A. Gopalan and A. Lawrence, "Calculating Power Dissipation for a H-Bridge or Half Bridge Driver." 2012.