JCBI

Journal of Computing & Biomedical Informatics
ISSN: 2710 - 1606

Volume 05 Issue 02
2023

Research Article
https://doi.org/10.56979/502/2023

Fault Detection in Rotors of Quadcopter UAV Using Convolutional Neural Network

Maria Panhwar[1*], Arbab Nighat[2], Farida Memon[2], and Shadab Kalhoro[3]

[1]Department of Mechatronics Engineering, Mehran University of Engineering and Technology Jamshoro, 76062, Pakistan.
[2]Department of Electronics Engineering, Mehran University of Engineering and Technology Jamshoro, 76062, Pakistan.
[3]Faculty of Information & Communication Technology, Universiti Tunku Abdul Rahman 31900, Kampar, Perak, Malaysia.
[*]Corresponding Author: Maria Panhwar. Email: mariapanhwar13@gmail.com.

Received: June 05, 2023 **Accepted:** August 29, 2023 **Published:** September 17, 2023.

Abstract: Mostly fault in quadcopter occurs in its rotors due to numerous reasons which leads towards flight failure. And due to that speedy fault detection is of great significance in practical applications. This work proposes a new fault detection method based on Convolutional Neural Networks (CNNs) on IMU sensors data to detect and classify the faulty and nominal flight conditions. Induced failure are introduced in quadcopter rotor section by varying the accelerometer such that yaw, pitch and roll data in Simulink/MATLAB. The data of the faulty model later on is used as input for the CNN learning model on WEKA platform. The proposed architecture is capable of automatically learning features and parameters which lead towards failure of the quadcopter from the supplied data as well as analyzing spatial and temporal fluctuations. On simulated data, empirical results demonstrate that our solution can classify various rotor fault types with an accuracy of 94%.

Keywords: Rotors failure; UAV, Convolutional Neural Network (CNN); Simulink; Fault induction; Fault identification.

1. Introduction

With the development of modern technology, UAV is drawing more attention because of its advantages of minimum cost and greater efficiency. Its safety and reliability problems are therefore raised. As the principal component of the flight control system, once the rotor fails, the UAV will inevitably be put in a dangerous situation. Therefore, conducting real-time fault analysis is important for the rotor so that timely actions can be taken to alleviate the fault influence and guarantee the flight safety. Many theories based on fault identification and diagnosis are proposed. The identification is based on vibrations. Whenever the unbalancing occurs due to changing thrust in air, the vibration occurs and through usage of some sensors or controlling units one can identify the fault. However, it is still not easy to identify and diagnose within the limit of time. Numerous methods and techniques are practiced locating the cause; additional sensors, introducing updated controllers and many more.

In recent years, data-driven approaches are getting more attention due to their robustness and reliability in fault detection. The fault detection based on a data-driven approach extracts various features from the original data and feeds them into the neural network model to obtain the fault detection results directly. The learning-based approach based on supervised learning requires experimental data containing the behavior of faulty quadrotors for training and labeling the fault cases. On the other hand, data-driven techniques based on Artificial Intelligence (AI) are being utilized with smartphones and driverless cars to address many complicated challenges. In AI machine learning methods and techniques are tested for UAVs, numerous filters and classifications have been carried out such as SVM and K-filters. Neural networks are the critical learning model in machine learning algorithms, which are applied in various application scenarios. Many types of neural networks have become mainstream in the data-driven based methods on UAV flight data fault detection and recovery.

This research proposes Convolutional Neural Network (CNN) for detection of faults. If an abnormal behavior is found in the sensor data, then the fault identification is performed. A model-based design is approached in this research work. The UAV tool included in MATLAB/Simulink is utilized to accomplish the needed aim. It offers several quadcopter functions and saves a lot of time. The variation is carried out in PID controller integrally, the fault is introduced in two ways 1. In environmental conditions such as wind speed, thrust and 2. In altitude controller that is yaw, pitch, roll and their angles with respect to other parameters. Later, the data acquired from Simulink model is used as input for neural network model. Weka platform is used for preprocessing and designing CNN model of our UAV quadcopter.

2. Literature Review

In the article [4] it is suggested that most of the current research on fault detection and identification focuses on problems in the UAV's sensors and actuators. When an accurate physical mechanism can be modelled, the UAV actuator failure detection methods based on physical models are extremely effective and have quick responses. And in article [6] the context of practical application, it is challenging to be satisfied. Due to their superior performance and adaptability, machine learning-based techniques have grown more popular. In article [7] through the appropriate mapping connection between output and input, machine learning methods may analyze measurement data and efficiently get the desired information. The approach is very simple to use and doesn't call for a lot of technical expertise. In machine learning algorithms, neural networks are the key learning model [2,3].

The Neural networks is the critical learning model in machine learning algorithms, which are applied in various application scenarios. Many types of neural networks have become mainstream in the data-driven based methods on UAV flight data fault detection and recovery [1]. For instance, improved the accuracy of UAV fault detection through the de-rending and de-noising algorithm [8]. proposed a long short-term memory and residual filtering (LSTM-RF) model to promote the performance on fault detection and recovery for the UAV sensors [9]. combined two-dimensional convolutional neural network (2DCNN) with long short-term memory (LSTM) to detect the insidious damages [10]. completed the detection of abnormal operating conditions of steam drums using the gate recurrent unit auto encoder neural network (GRU-Auto Encoder) [11]. designed a multi-channel convolutional neural network neural network to achieve the data classification [12]. proposed a seismic facies classification method based on the convolutional neural network (CNN) to improve classification accuracy, in which the CNN was used for feature extraction of image data [13]. The above literature shows a good performance on fault detection and feature learning. However, they have a weakness on the feature selection and fault detection effected by the irrelevant data in time series sequences. the exhibition examination of a completely tuned brain network prepared with the lengthy negligible asset dispensing network (EMRAN) calculation for ongoing recognizable proof of a quadcopter. Outspread premise capability organization (RBF) in view of framework recognizable proof can be used as an elective strategy for quadcopter displaying. To forestall the neurons and organization boundaries choice difficulty during experimentation approach, RBF with EMRAN preparing calculation is proposed.

This programmed tuning calculation will execute the organization developing and pruning strategy to add or take out neurons in the RBF. The EMRAN's performance is contrasted and the insignificant asset dispensing network (MRAN) preparing for 1000 information yields undeveloped demeanor information. The discoveries show that the EMRAN technique produces a quicker mean preparation season of generally 4.16 ms for neuron size of up to 88 units contrasted with MRAN at 5.89 ms with a slight decrease in expectation precision. [15]. In article [16] the model purposes a cutting-edge CNN, DenseNet-161, to order the pictures extricated from a forward-looking camera of a UAV. Considering grouping result, the model creates an important control order to securely explore the UAV in an obscure hallway climate. Rather than a portion of the past techniques, framework just purposes the contributions from a monocular camera.

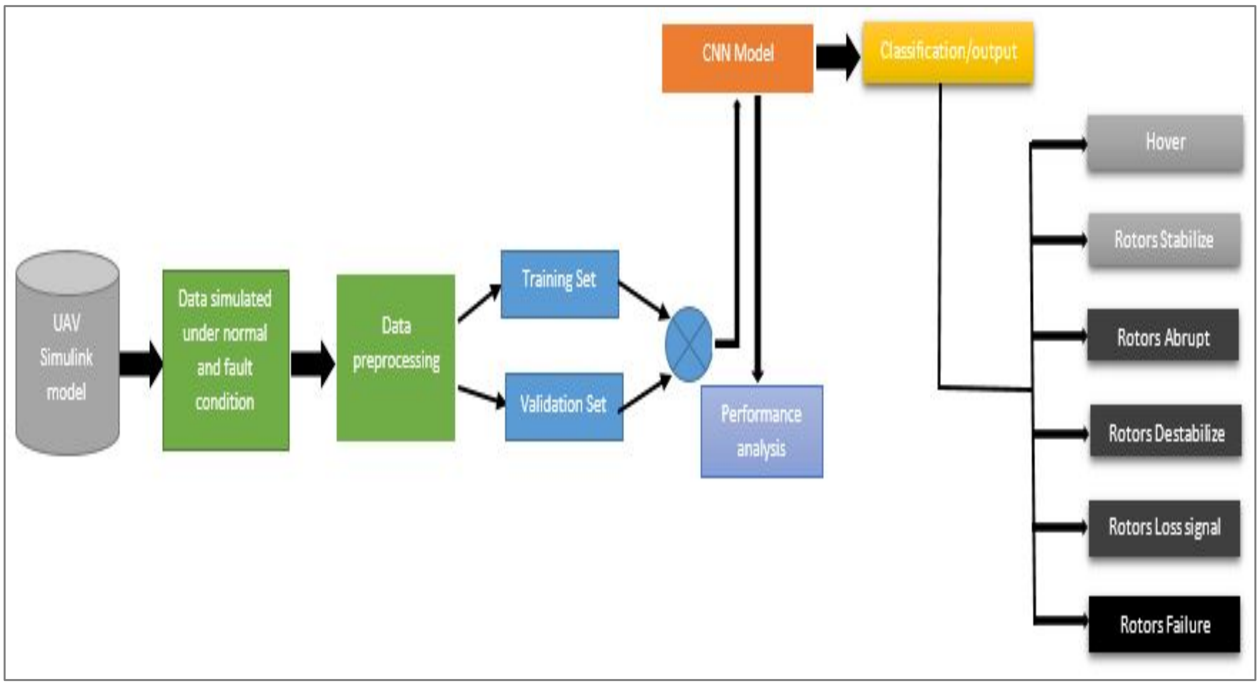
Thus, it is computationally extremely productive, likewise presenting a dataset of pictures relating to various places of the UAV all through various lengths of hallways, with shifting aspects. Likewise, the aftereffect of our route calculation in certifiable hallway conditions is exceptionally reassuring. Whereas in [17], This article presents a clever methodology for identifying and detaching broken actuators in exceptionally excess Multirotor UAVs utilizing flowed Profound Brain Organization (DNN) models. The proposed Issue Identification and Separation (FDI) structure joins Long Transient Memory (LSTM)- based shortcoming discovery and broken actuator finder models to accomplish ongoing checking. The review centers around a Hexa rotor multirotor UAV furnished with sixteen rotors. To handle the intricacy of FDI coming about because of overt repetitiveness, a parceling method is presented in view of framework elements. The proposed FDI conspire is made from a locale classifier model liable for distinguishing shortcomings and issue finder models that definitively decide the area of the bombed actuator. Broad preparation and testing of the models show high precision, with the provincial classifier model accomplishing 98.97% exactness and the shortcoming finder model accomplishing 99.107% precision. Besides, the plan was coordinated into the flight control arrangement of the UAV, prior to being tried by means of both continuous observing in the reproduction climate and examination of recorded genuine flight information. The models display amazing execution in recognizing and limiting infused shortcomings. Thusly, utilizing DNN models and the dividing strategy, this examination offers a promising technique for precisely identifying and disconnecting broken actuators, subsequently working on the general execution and constancy of profoundly repetitive Multirotor UAVs in different functional situations.

In this research, we propose CNN deep learning architecture to detect and identify the actuator fault from drone's IMU sensor data.

3. Proposed Solution & Methods

In this study, our aim to develop a fault classifier for UAV quadcopter using simulated data. Figure 1 illustrates the proposed methodology of our work. A Simulink model of a quadcopter is taken and is used to generate data under both normal and specified fault conditions. The data collected from Simulink model under nominal and abrupt conditions by varying the environmental conditions such that wind speed and wind angles and its impact on accelerometer parameters that are yaw, pitch and roll values. The generated data from the Simulink model preprocessed and divided for training and validation with these training samples under supervised learning the CNN based classifier is developed to classify rotor's fault. Finally, performance analysis of designed model is carried out which evaluate the final predicated and actual results of our model and provide the accuracy of our model.

Figure 1. Block diagram of CNN based UAV actuator fault detection approach.



3.1 UAV-Simulink model

The Simulink model of UAV quadrotor is shown in Fig.2. It consists of three subsystems, i.e., on board sensors, multirotor and external sensors. Multirotor subsystem contains three controllers for position, altitude, and yaw. To acquire data from quadcopter sensors, first input signals are applied in safe range to identify the stability of quadcopter. After collecting the data of normal flight of quadcopter, we induced the fault in Simulink model of UAV by alternating the PID controller data.

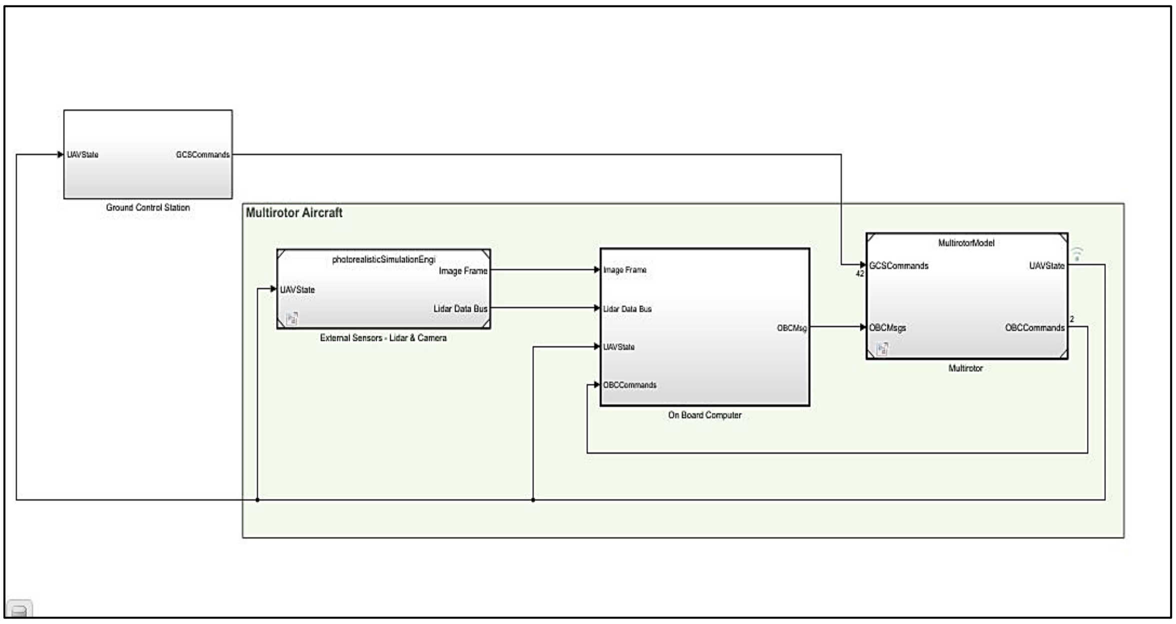


Figure 2. Simulink model of UAV quadcopter

3.2 Data simulated under normal and fault condition.

An efficient and secure method is to train a defect classifier with simulated data and use it on a real quadrotor. The major concern of this study is to detect the actuator fault based heading and altitude parameters i.e., yaw, pitch and roll measured through IMU sensor. First, a step input signal of 0.5mV to 5V is applied in safe range to identify the stability of quadcopter model. The fault is then introduced in yaw, pitch and roll by varying their angles in heading and position as shown in Fig.3. Besides this, we also increase the average wind speed using onboard sensors and recorded the variation in the roll parameter. From the PID controller shown in Fig.3. In figure 3 the data related to heading, position as well as altitude controller is taken for the classifier.

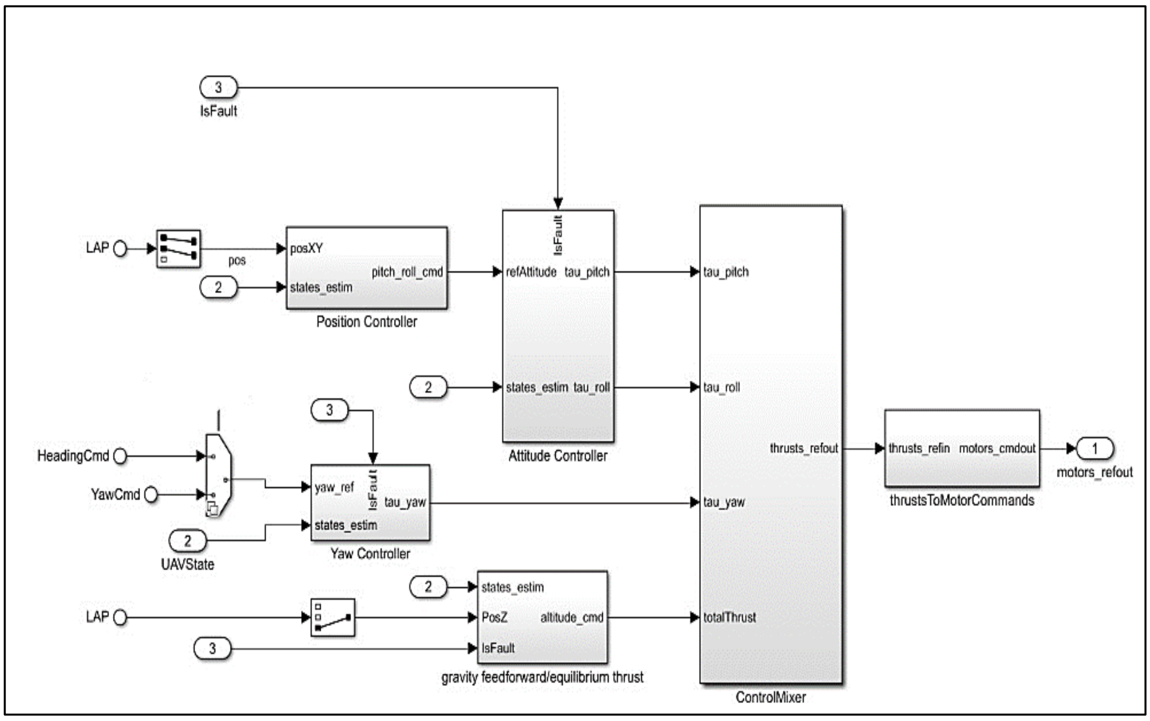


Figure 3. PID controller of UAV quadcopter in Simulink

For the safe flight all the gain parameter of pitch, roll and yaw are kept under the safe range as shown in Table1.

Table 1. Gain parameters of PID controller before fault injection

S.No.	Model description	Gain parameters
1	Outer proportional loop control for pitch	Kp =8.5
2	Inner PI control for pitch angular velocity	Kp = Ki =0.04
3	Outer proportional loop control for roll is	Kp =8.57
4	Inner PI control for roll angular velocity	Kp = Ki =0.04
5	PID control for yaw	Kp = Ki = Kd =4.28

After acquiring the data of safe flight under controllable parameters, we introduced fault in Simulink model by varying its IMU accelerometer data, i.e., yaw, pitch, and roll. The fault is injected in multirotor as shown in Fig.4.

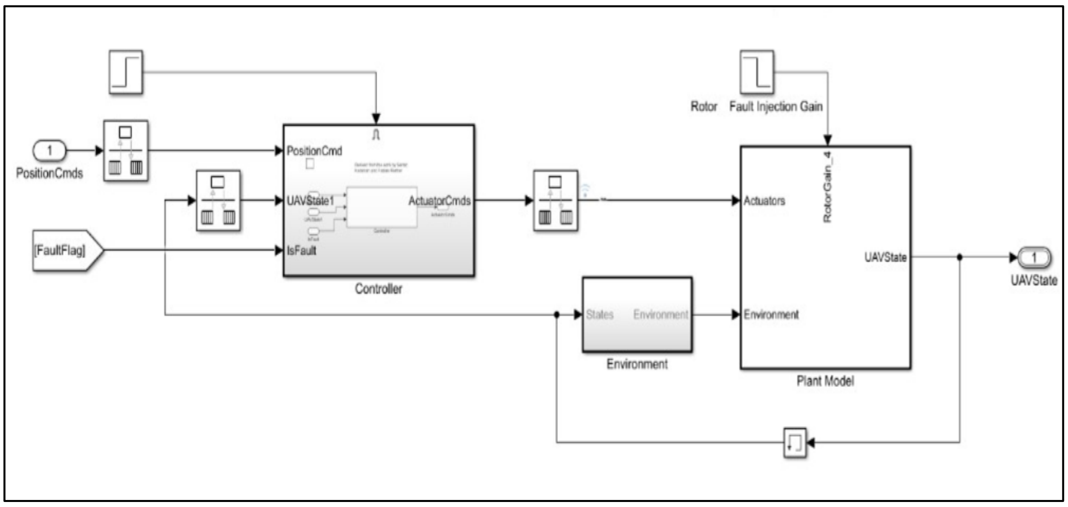


Figure 4. Injecting fault in multirotor.

The fault is injected in plant model through variations in PID parameters of multirotor system. Through the increase and decrease in gain parameters, variations in yaw, roll and pitch are recorded. A set of data is collected for 100 times flight run. The data is collected at numerous stances for yaw, pitch, and roll. The gain parameters of yaw, pitch and roll after injection of the fault are given in the Table2.

Table 2. Gain parameters of PID controller after fault injection

S.no:	Model description	Gain Parameters after Injecting fault
1	Outer proportional loop control for pitch	Kp=13.98
2	Inner PI control for pitch angular velocity	Kp=0.069 ,Ki=0.05
3	Outer proportional loop control for roll is	*Kp*=13.87
4	Inner PI control for roll angular velocity	*Kp*=0.52,*Ki*=0.03
5	PID control for yaw	Kp=2.56,Ki=0.06,Kd=6.9

3.3 CNN Based Fault Classifier

CNNs, also referred to as convolutional neural networks, are an effective kind of neural network used for processing and identifying tasks. It contains layers including convolutional, activation, and pooling layers. In this study, the actuator problems are categorized using CNN architecture. The CNN model outputs with the label utilizing the data obtained from the Simulink model as input. The CNN architecture for the suggested job is shown in Fig. 5, which includes many layers and an output layer. The ultimate output of a CNN is created by the output layer, which often results in a probability distribution of classes for a classification problem. The output layer in this case uses a hyperbolic tangent activation function to create a probability distribution over the various classes. The deep learning model designed is trained with six classes as different types of rotor condition i.e hover, rotor stabilize, rotor abrupt, rotor destabilize, rotor loss signal and rotor failure. The predicted class for the provided data is determined based on the highest possibility class and then categorized as one of specified rotor condition using the output layer

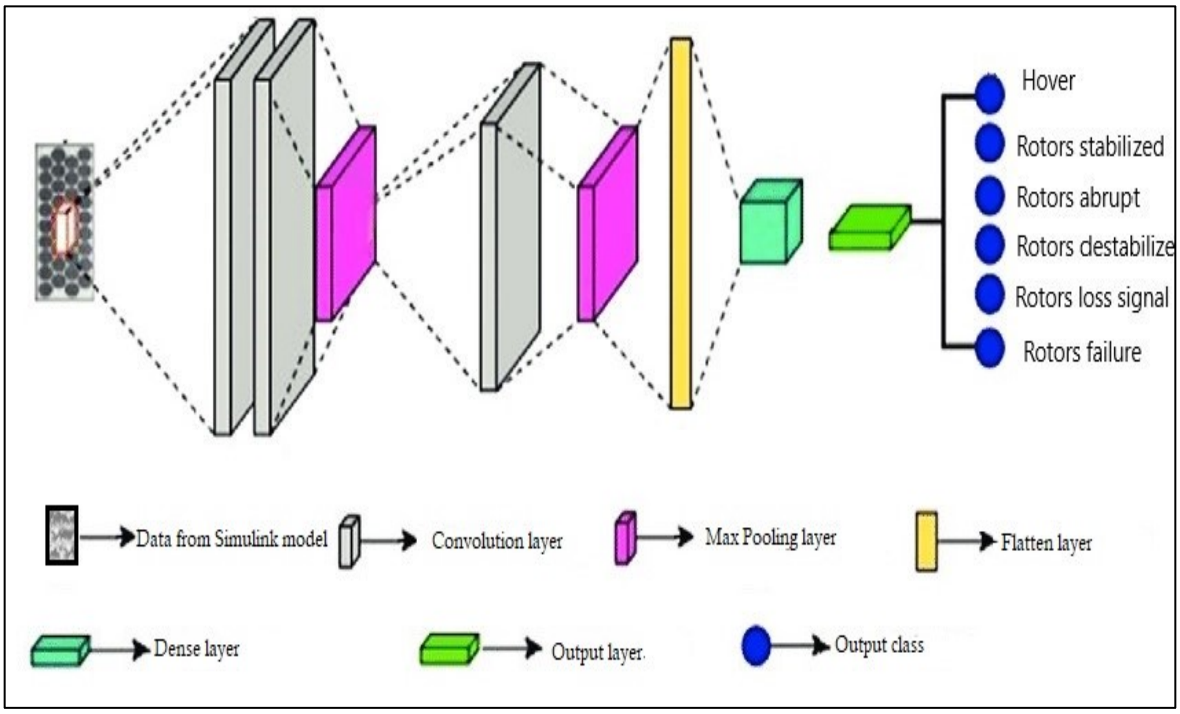
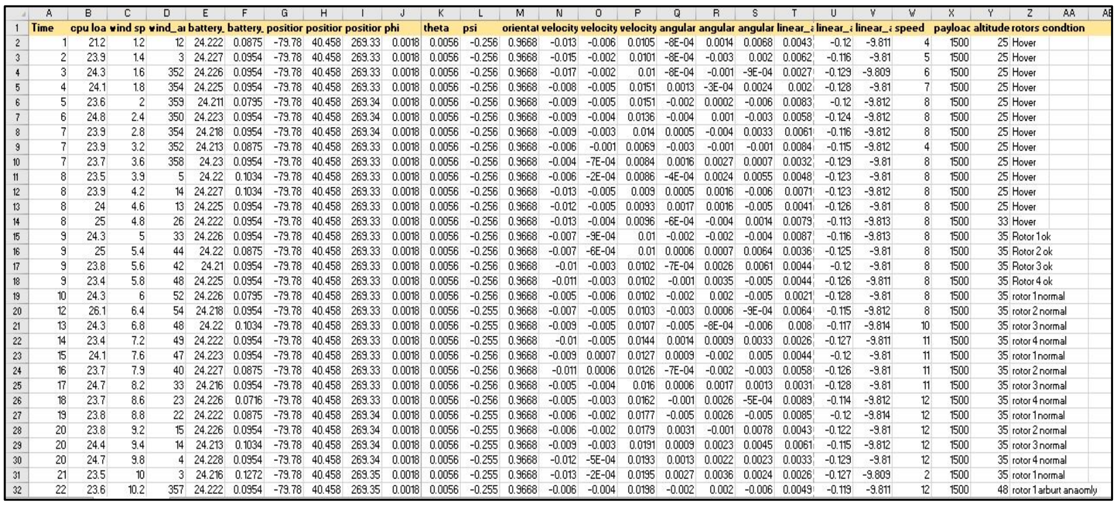


Figure 5. Proposed CNN model for actuator fault classification

4. Results and Discussions

4.1 Dataset collection

The dataset for this study is developed from the Simulink model. The acquired data is converted first into .mat file and then into .csv file. The collected dataset contains 26 attributes with 3000 entries having numeric, string and binary values. Data preprocessing is used to turn the original data into a clean data set. The dataset is collected in raw format, which makes analysis impractical. We perform various operations including data discretization and filtering. The resulting data is then labelled to show six rotor conditions. The labelled dataset was split into two sets: a training set with 70% of the data and a validation set with 30% of the data. Some of the rows of the dataset are shown in figure 6 where the last column indicates the label of the rotor condition.



	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	AA
1	Time	cpu loa	wind sp	wind_a	battery_	battery_	positior	positior	positior	phi	theta	psi	oriental	velocity	velocity	velocity	angular	angular	angular	linear_a	linear_a	linear_a	speed	payloac	altitude	rotors condtion	
2	1	21.2	1.2	12	24.222	0.0875	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.013	-0.006	0.0105	-8E-04	0.0014	0.0068	0.0043	-0.12	-9.811	4	1500	25	Hover	
3	2	23.9	1.4	3	24.227	0.0954	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.015	-0.002	0.0101	-8E-04	-0.003	0.002	0.0062	-0.116	-9.81	5	1500	25	Hover	
4	3	24.3	1.6	352	24.226	0.0954	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.017	-0.002	0.01	-8E-04	-0.001	-9E-04	0.0027	-0.129	-9.809	6	1500	25	Hover	
5	4	24.1	1.8	354	24.225	0.0954	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.008	-0.005	0.0151	0.0013	-3E-04	0.0024	0.002	-0.128	-9.81	7	1500	25	Hover	
6	5	23.6	2	359	24.211	0.0795	-79.78	40.458	269.34	0.0018	0.0056	-0.256	0.9668	-0.009	-0.005	0.0151	-0.002	0.0002	-0.006	0.0083	-0.12	-9.812	8	1500	25	Hover	
7	6	24.8	2.4	350	24.223	0.0954	-79.78	40.458	269.34	0.0018	0.0056	-0.256	0.9668	-0.009	-0.004	0.0136	-0.004	0.001	-0.003	0.0058	-0.124	-9.812	8	1500	25	Hover	
8	7	23.9	2.8	354	24.218	0.0954	-79.78	40.458	269.34	0.0018	0.0056	-0.256	0.9668	-0.009	-0.003	0.014	0.0005	-0.004	0.0033	0.0061	-0.116	-9.812	8	1500	25	Hover	
9	7	23.9	3.2	352	24.213	0.0875	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.006	-0.001	0.0069	-0.003	-0.001	-0.001	0.0084	-0.115	-9.812	4	1500	25	Hover	
10	7	23.7	3.6	358	24.23	0.0954	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.004	-7E-04	0.0084	0.0016	0.0027	0.0007	0.0032	-0.129	-9.81	8	1500	25	Hover	
11	8	23.5	3.9	5	24.22	0.1034	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.006	-2E-04	0.0086	-4E-04	0.0024	0.0055	0.0048	-0.123	-9.81	8	1500	25	Hover	
12	8	23.9	4.2	14	24.227	0.1034	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.013	-0.005	0.009	0.0005	0.0016	-0.006	0.0071	-0.123	-9.812	8	1500	25	Hover	
13	8	24	4.6	13	24.225	0.0954	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.012	-0.005	0.0093	0.0017	0.0016	-0.005	0.0041	-0.126	-9.81	8	1500	25	Hover	
14	8	25	4.8	26	24.222	0.0954	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.013	-0.004	0.0096	-6E-04	-0.004	0.0014	0.0079	-0.113	-9.813	8	1500	33	Hover	
15	9	24.3	5	33	24.226	0.0954	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.007	-9E-04	0.01	-0.002	-0.002	-0.004	0.0087	-0.116	-9.813	8	1500	35	Rotor 1 ok	
16	9	25	5.4	44	24.22	0.0875	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.007	-6E-04	0.01	0.0006	0.0007	0.0064	0.0036	-0.125	-9.81	8	1500	35	Rotor 2 ok	
17	9	23.8	5.6	42	24.21	0.0954	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.01	-0.003	0.0102	-7E-04	0.0026	0.0061	0.0044	-0.12	-9.81	8	1500	35	Rotor 3 ok	
18	9	23.4	5.8	48	24.225	0.0954	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.011	-0.003	0.0102	-0.001	0.0035	-0.005	0.0044	-0.126	-9.811	8	1500	35	Rotor 4 ok	
19	10	24.3	6	52	24.226	0.0795	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.005	-0.006	0.0102	-0.002	0.002	-0.005	0.0021	-0.128	-9.81	8	1500	35	rotor 1 normal	
20	12	26.1	6.4	54	24.218	0.0954	-79.78	40.458	269.33	0.0018	0.0056	-0.255	0.9668	-0.007	-0.005	0.0103	-0.003	0.0006	-9E-04	0.0064	-0.115	-9.812	8	1500	35	rotor 2 normal	
21	13	24.3	6.8	48	24.22	0.1034	-79.78	40.458	269.33	0.0018	0.0056	-0.255	0.9668	-0.009	-0.005	0.0107	-0.005	-8E-04	-0.006	0.008	-0.117	-9.814	10	1500	35	rotor 3 normal	
22	14	23.4	7.2	49	24.222	0.0954	-79.78	40.458	269.33	0.0018	0.0056	-0.255	0.9668	-0.01	-0.005	0.0144	0.0014	0.0009	0.0033	0.0026	-0.127	-9.811	11	1500	35	rotor 4 normal	
23	15	24.1	7.6	47	24.223	0.0954	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.009	0.0007	0.0127	0.0009	-0.002	0.005	0.0044	-0.12	-9.81	11	1500	35	rotor 1 normal	
24	16	23.7	7.9	40	24.227	0.0875	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.011	0.0006	0.0126	-7E-04	-0.002	-0.003	0.0058	-0.126	-9.81	11	1500	35	rotor 2 normal	
25	17	24.7	8.2	33	24.216	0.0954	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.005	-0.004	0.016	0.0006	0.0017	0.0013	0.0031	-0.128	-9.81	11	1500	35	rotor 3 normal	
26	18	23.7	8.6	23	24.226	0.0716	-79.78	40.458	269.33	0.0018	0.0056	-0.256	0.9668	-0.005	-0.003	0.0162	-0.001	0.0026	-5E-04	0.0089	-0.114	-9.812	12	1500	35	rotor 4 normal	
27	19	23.8	8.8	22	24.222	0.0875	-79.78	40.458	269.34	0.0018	0.0056	-0.255	0.9668	-0.006	-0.002	0.0177	-0.005	0.0026	-0.005	0.0085	-0.12	-9.814	12	1500	35	rotor 1 normal	
28	20	23.8	9.2	15	24.226	0.0954	-79.78	40.458	269.34	0.0018	0.0056	-0.255	0.9668	-0.006	-0.002	0.0179	0.0031	-0.001	0.0078	0.0043	-0.122	-9.81	12	1500	35	rotor 2 normal	
29	20	24.4	9.4	14	24.213	0.1034	-79.78	40.458	269.34	0.0018	0.0056	-0.255	0.9668	-0.009	-0.003	0.0191	0.0009	0.0023	0.0045	0.0061	-0.115	-9.812	12	1500	35	rotor 3 normal	
30	20	24.7	9.8	4	24.228	0.0954	-79.78	40.458	269.34	0.0018	0.0056	-0.255	0.9668	-0.012	-5E-04	0.0193	0.0013	0.0022	0.0023	0.0033	-0.129	-9.81	12	1500	35	rotor 4 normal	
31	21	23.5	10	3	24.216	0.1272	-79.78	40.458	269.35	0.0018	0.0056	-0.255	0.9668	-0.013	-2E-04	0.0195	0.0027	0.0036	0.0024	0.0026	-0.127	-9.809	2	1500	35	rotor 1 normal	
32	22	23.6	10.2	357	24.222	0.0954	-79.78	40.458	269.35	0.0018	0.0056	-0.255	0.9668	-0.006	-0.004	0.0198	-0.002	0.002	-0.006	0.0049	-0.119	-9.811	12	1500	48	rotor 1 arburt anaomly	

Figure 6. Data collected from Simulink model.

4.2 Model design & analysis

For the categorization of rotor conditions labeled as follows: hover, rotor stabilize, rotor abrupt, rotor destabilize, rotor loss signal, and rotor failure, a CNN classification model is constructed and trained using the Weka platform. We sent the preprocessed data into the CNN model to train it, keeping 70% of the data as training data and the remaining 30% as validation data. The gradient descent optimizer is used to train the model using a learning rate of 0.04, a batch size of 100, and 300 epochs. The findings show that for these types of rotor defects, the created model achieves a greater level of classification accuracy. Table 4 lists the hyper-parameters used for the model training as well as the suggested model's validation accuracy.

Table 3. Selected hyper parameters.

S.No:	Parameters	Value
1	Total filters used	3
2	Kernel size for convolution	3x3 filter size
4	Activation function	Hyperbolic tangent
5	Optimizer	Gradient descent
6	Max epoch	300
7	Learning rate	0.04
8	Momentum	0.2
9	Batch size	100
10	Validation split	20

4.3. Confusion Matrix

In figure 07 confusion matrix gives an overview of the generated model's performance and demonstrates how well it can predict most of the classes. Analysis of the matrix reveals that the data of six rotor failure scenarios was effectively trained in our CNN model.

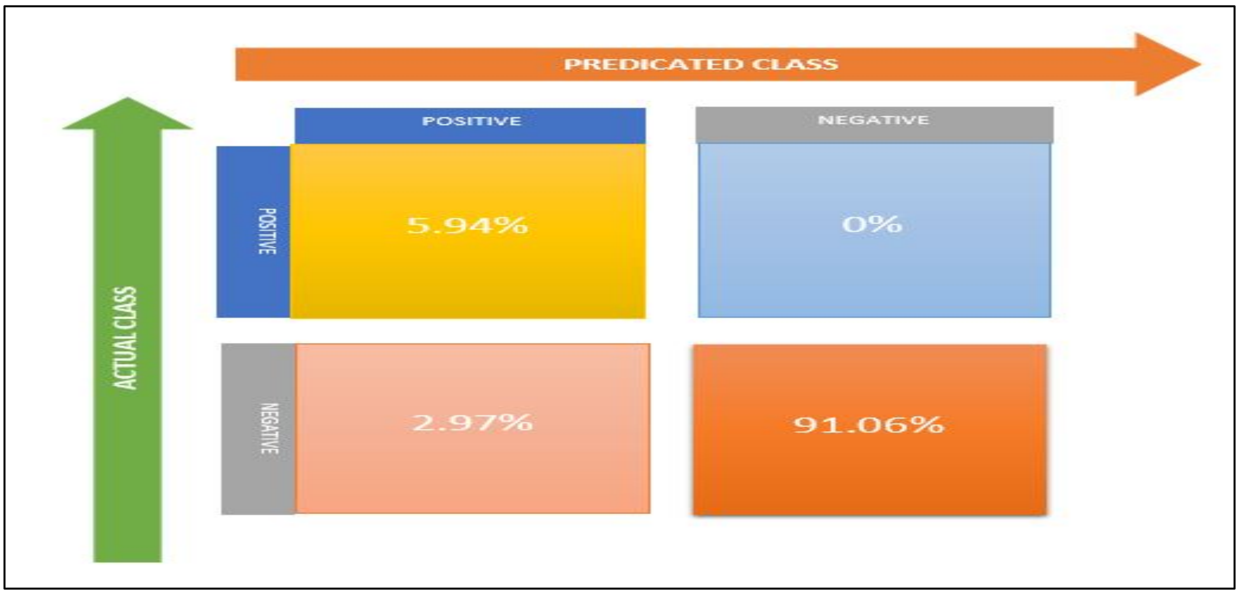


Figure 7a. Confusion Matrix

From Figure 7 (a) it is justified that those cases which have true positive values such as hover are predicated positive with 5.94%, whereas no false negative value is recognized which is 0%. The false positive is 2.97% and the true negative values which are labelled as failure cases data are correctly classified in true negative values with 91.06%. Similarly, the confusion matrix of other five conditions are carried out.

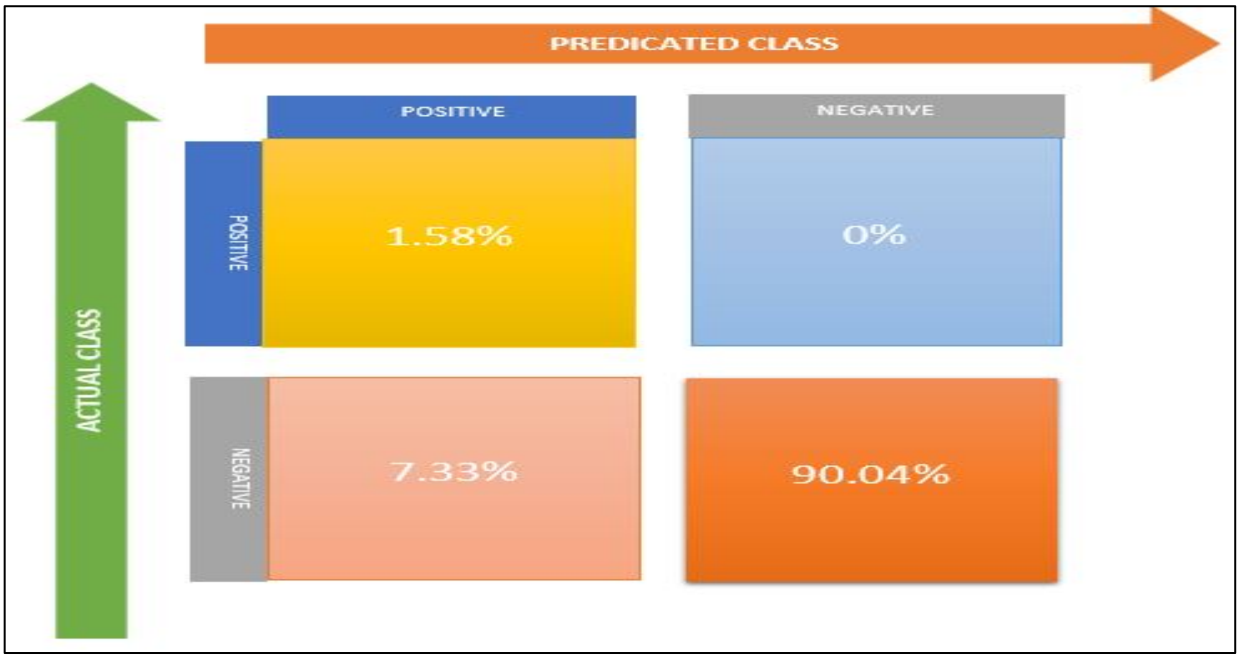


Figure 7b. Confusion Matrix

In Figure 7(b) it is justified that those cases which have true positive values for rotor stabilized are predicated positive with 1.58%, whereas no false negative value is recognized which is 0%. The false positive is 7.33% and the true negative values which are labelled as failure cases data are correctly classified in true negative values with 90.04%.

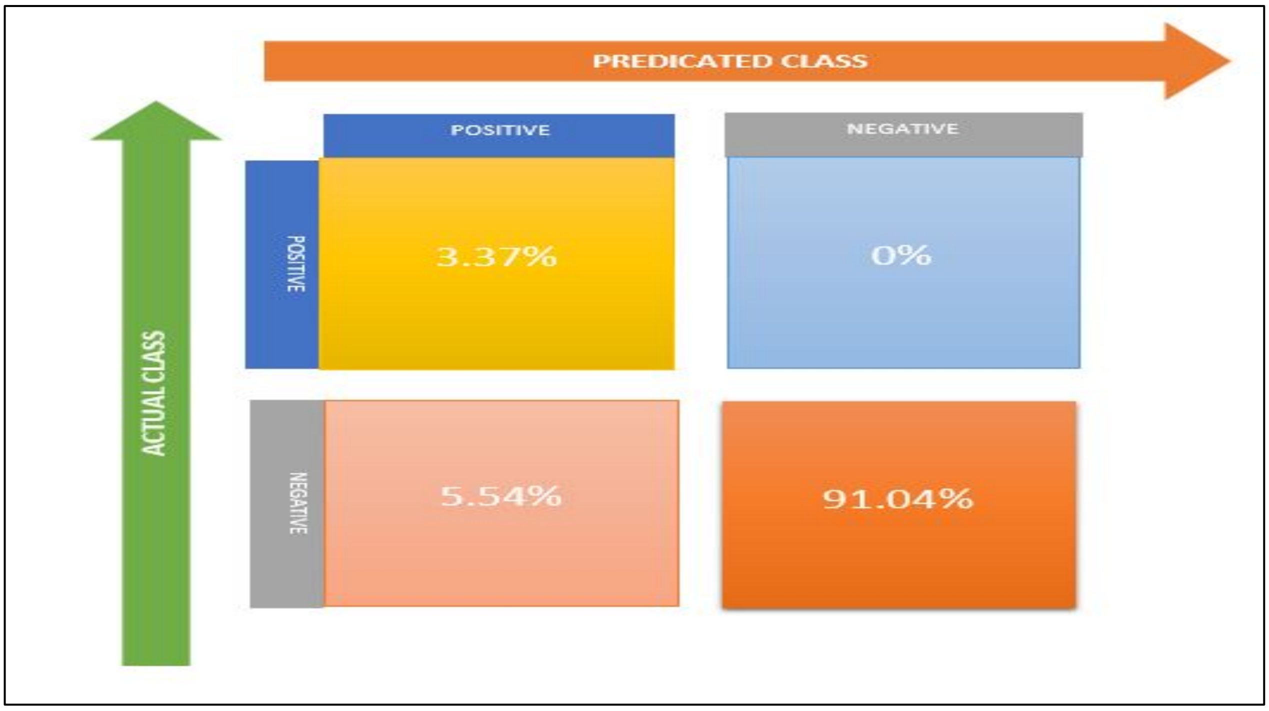


Figure 7c. Confusion Matrix

In Figure 7 (c) it is justified that those cases which have true positive values for rotor abrupt are predicated positive with 3.37%, whereas no false negative value is recognized which is 0%. The false positive is 5.54% and the true negative values which are labelled as failure cases data are correctly classified in true negative values with 91.04%.

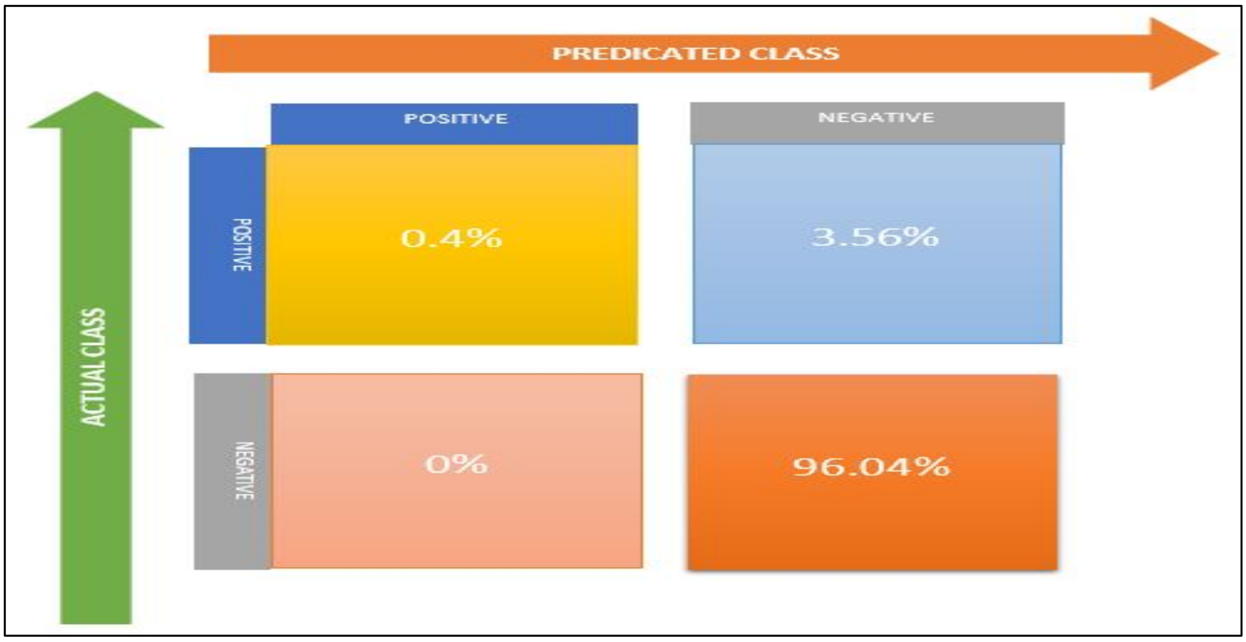


Figure 7d. Confusion Matrix

In Figure 7 (d) it is justified that those cases which have true positive values for rotor destabilized are predicated positive with 0.4%, whereas no false negative value is recognized which is 3.56%. The false positive is 0% and the true negative values which are labelled as failure cases data are correctly classified in true negative values with 96.04%.

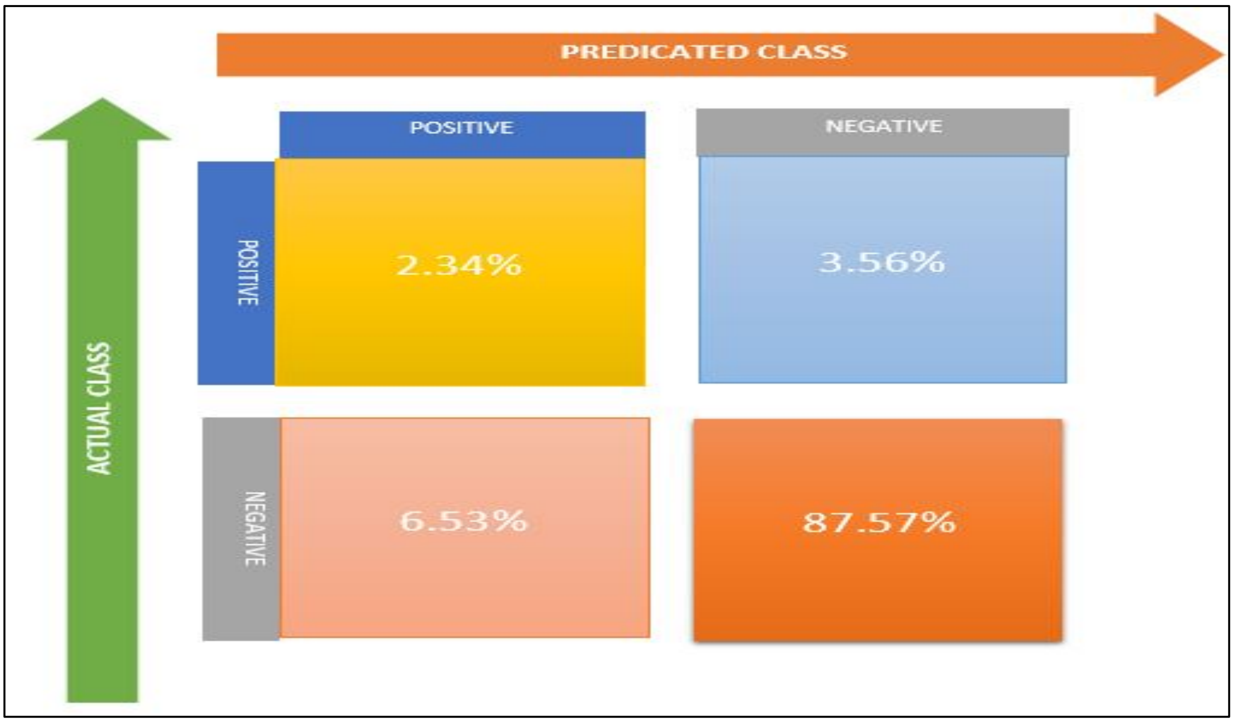


Figure 7e. Confusion Matrix

Figure 7 (e) it is justified that those cases which have true positive values for rotor loss signal are predicated positive with 2.34 %, whereas no false negative value is recognized which is 3.56%. The false positive is 6.53% and the true negative values which are labelled as failure cases data are correctly classified in true negative values with 87.57%.

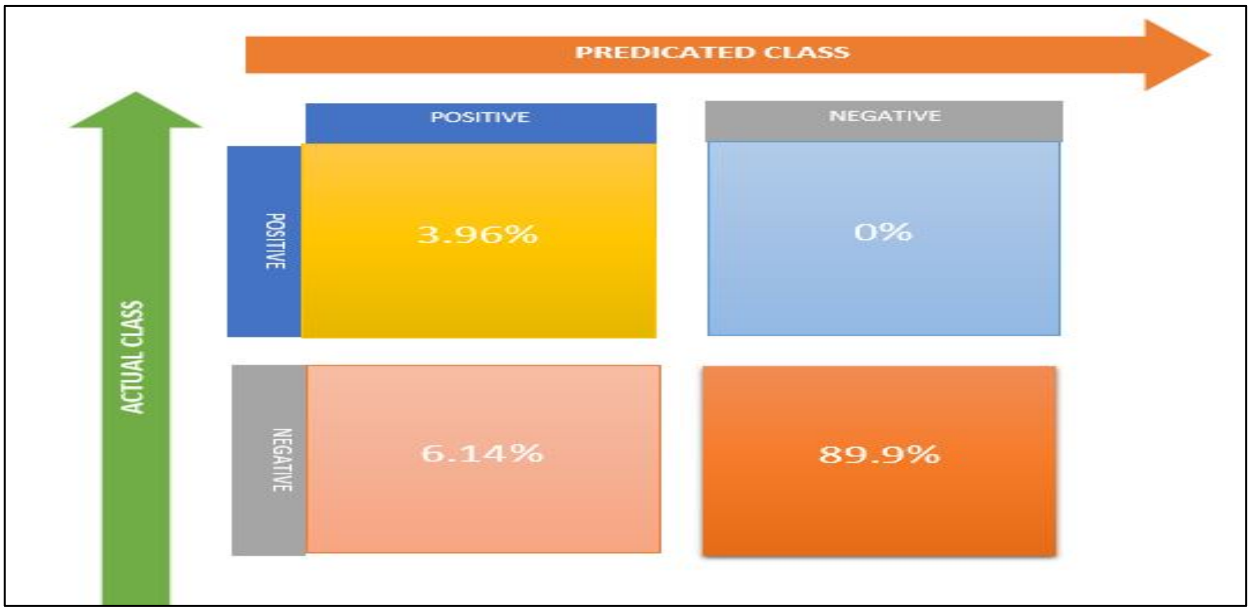


Figure 7f. Confusion Matrix

In Figure 7 (f) it is justified that those cases which have true positive values for rotor stabilized are predicated positive with 3.96%, whereas no false negative value is recognized which is 0%. The false positive is 6.14% and the true negative values which are labelled as failure cases data are correctly classified in true negative values with 89.9%.

4.4 Performance Report

The performance report shown in Figure 08 contains significant information regarding the accuracy, recall, and f-measure of the created model for each situation. Upon examining the report, we conclude that all rotors of quadcopter display precision rate in different scenarios as shown in Figure 9 of different ROCs.

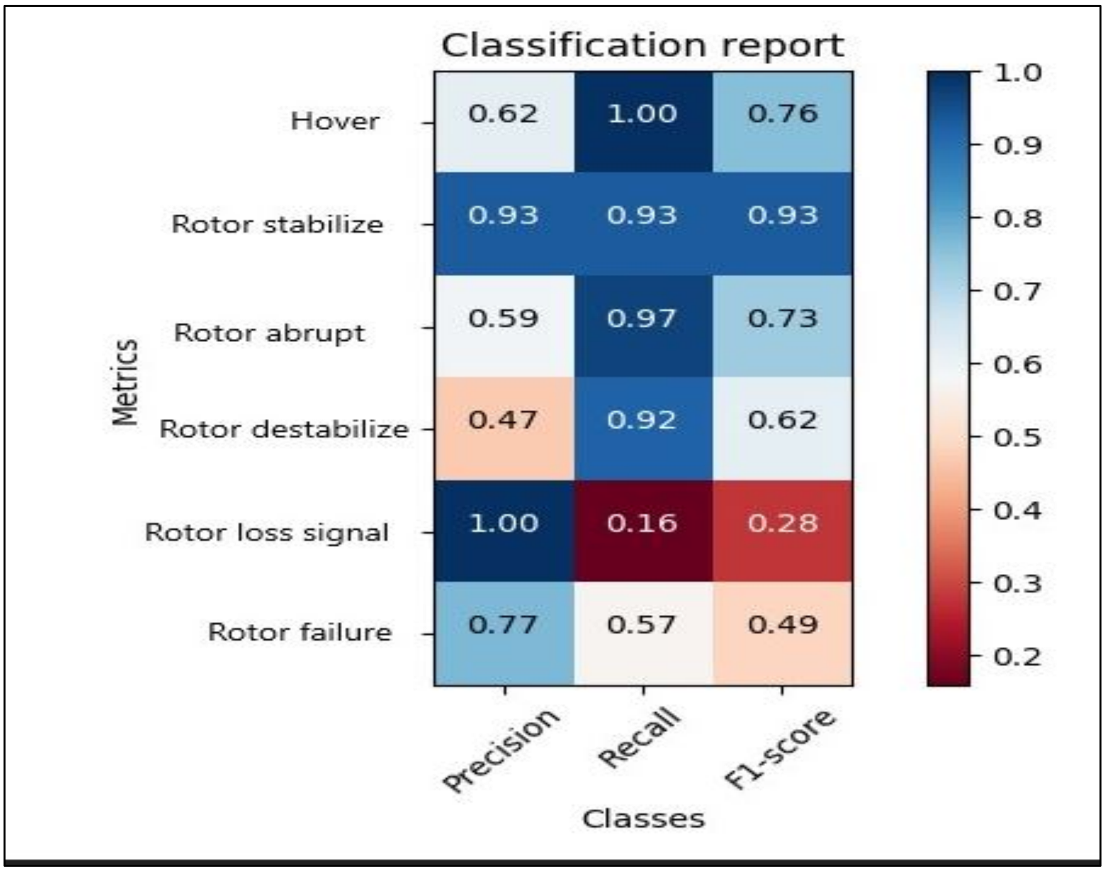


Figure 8. Showed the Developed model classification report.

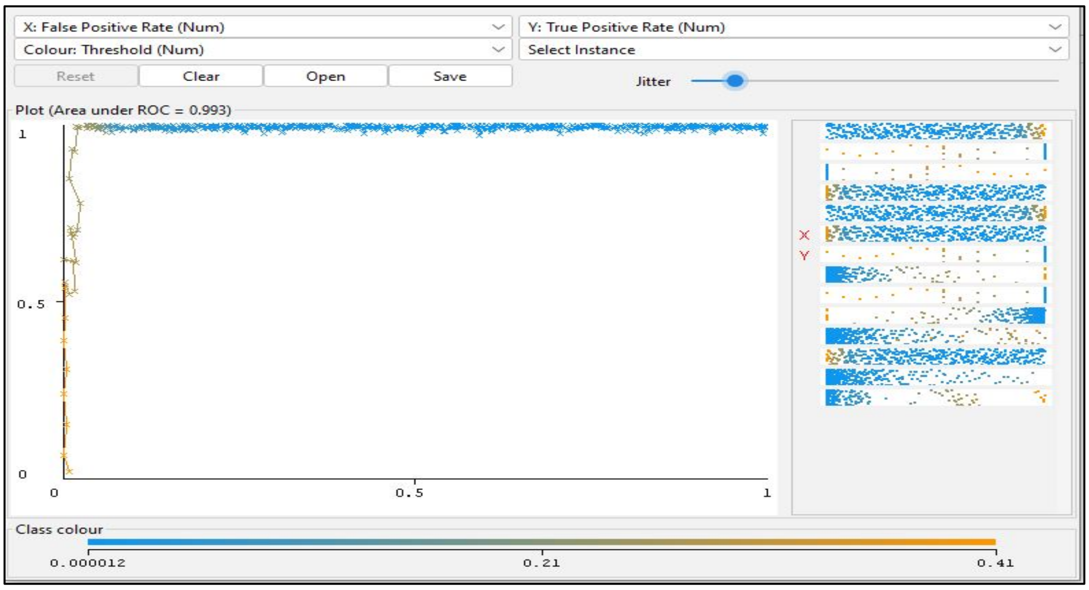


Figure 9a. ROC curve of Hover

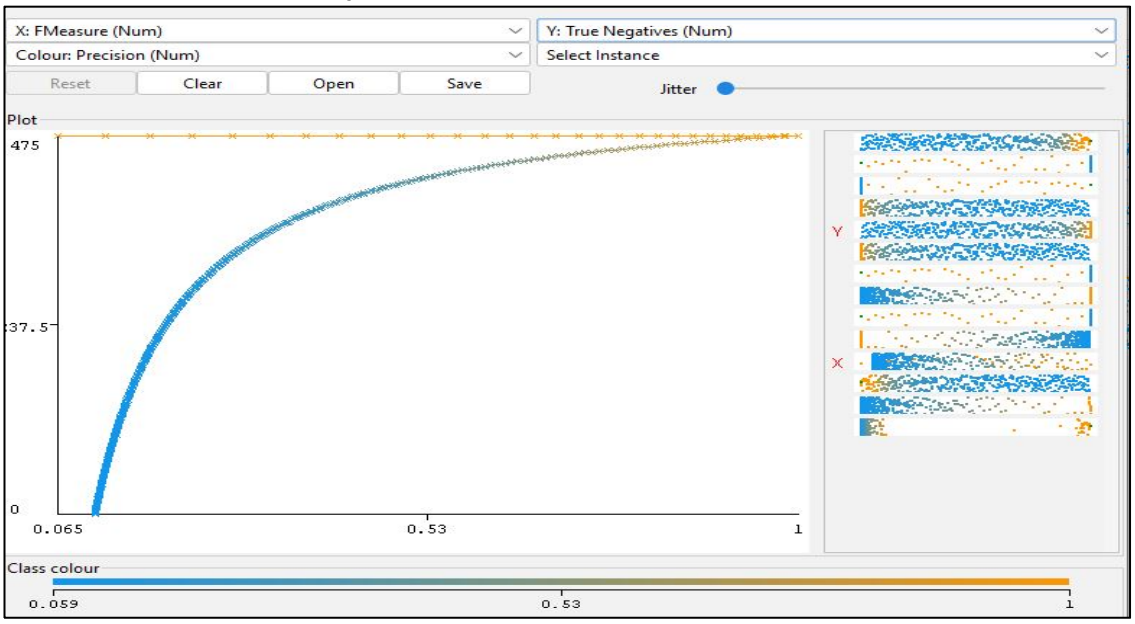


Figure 9b.ROC curve of Rotor stabilize.

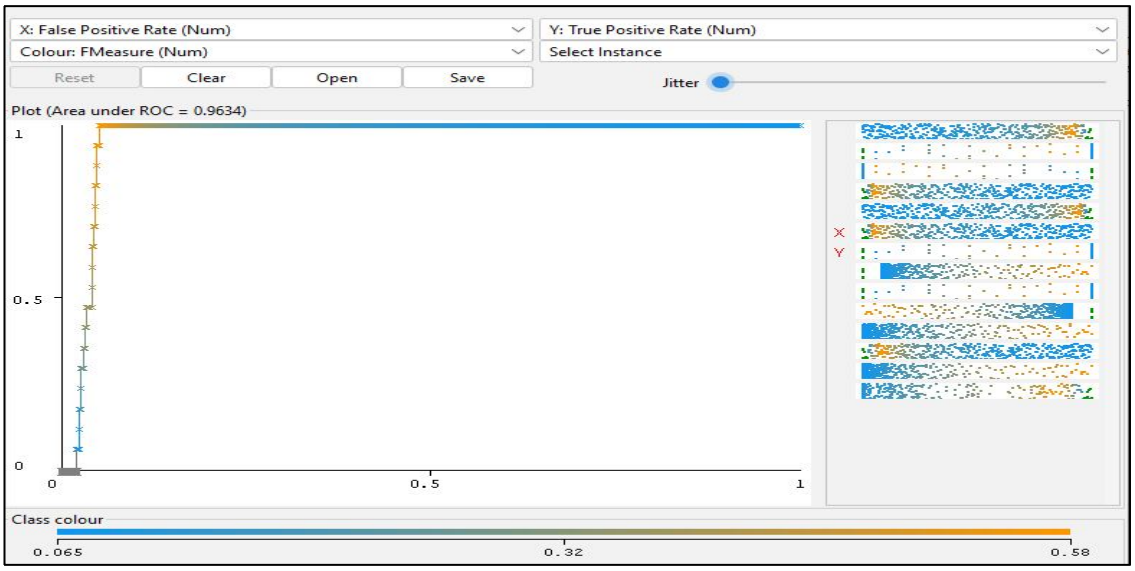


Figure 9c ROC curve of Rotor abrupt

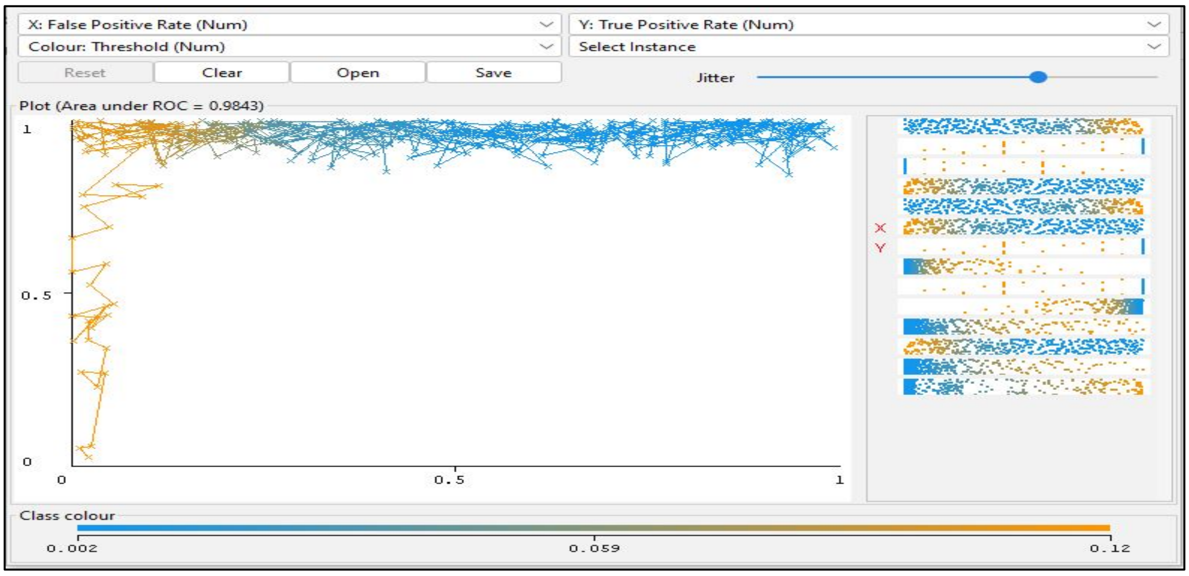


Figure 9d. ROC curve of Rotor destabilize.

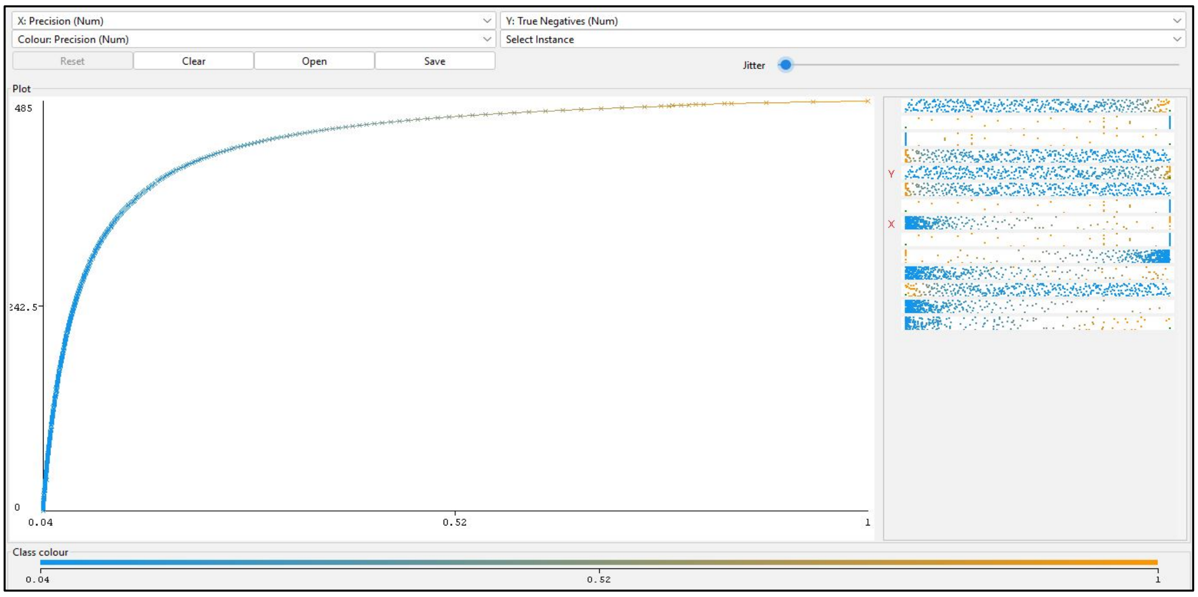


Figure 9e. ROC curve of Rotor loss signal

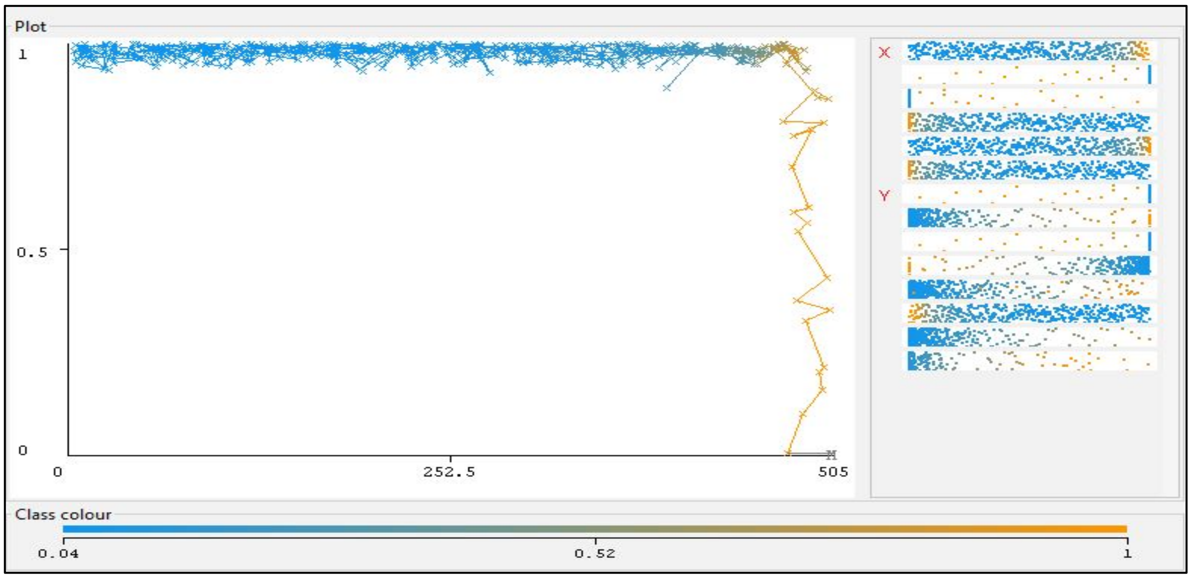
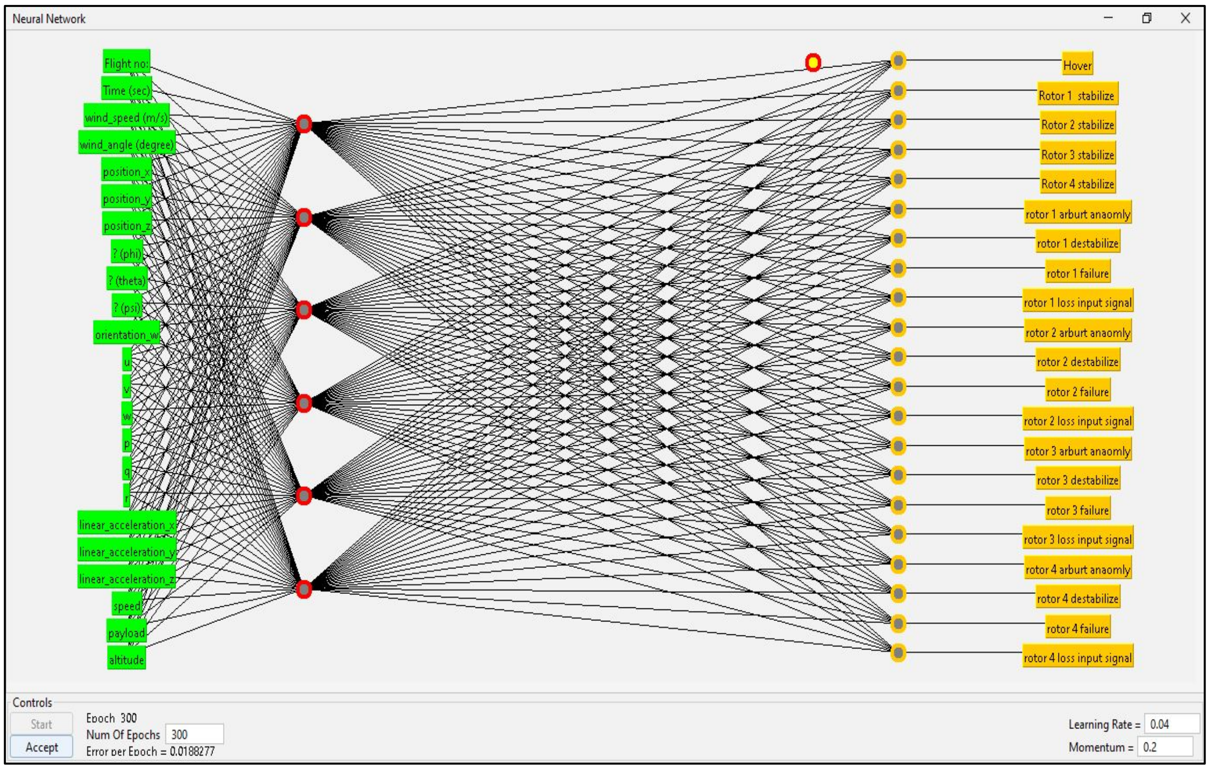


Figure 9f. ROC curve of Rotor failure

Figure 9. shows the ROC plots (a, b, c, d ,e ,f) for rotors. The ability of classifiers, i.e., how many classifiers can predict only samples of their class is described by ROC plots as shown in Figure 9. In this Fig (i.e., Fig. 9), the nearness of curves in ROC plot towards their top left corner indicates better performance.

4.5 Model Implementation

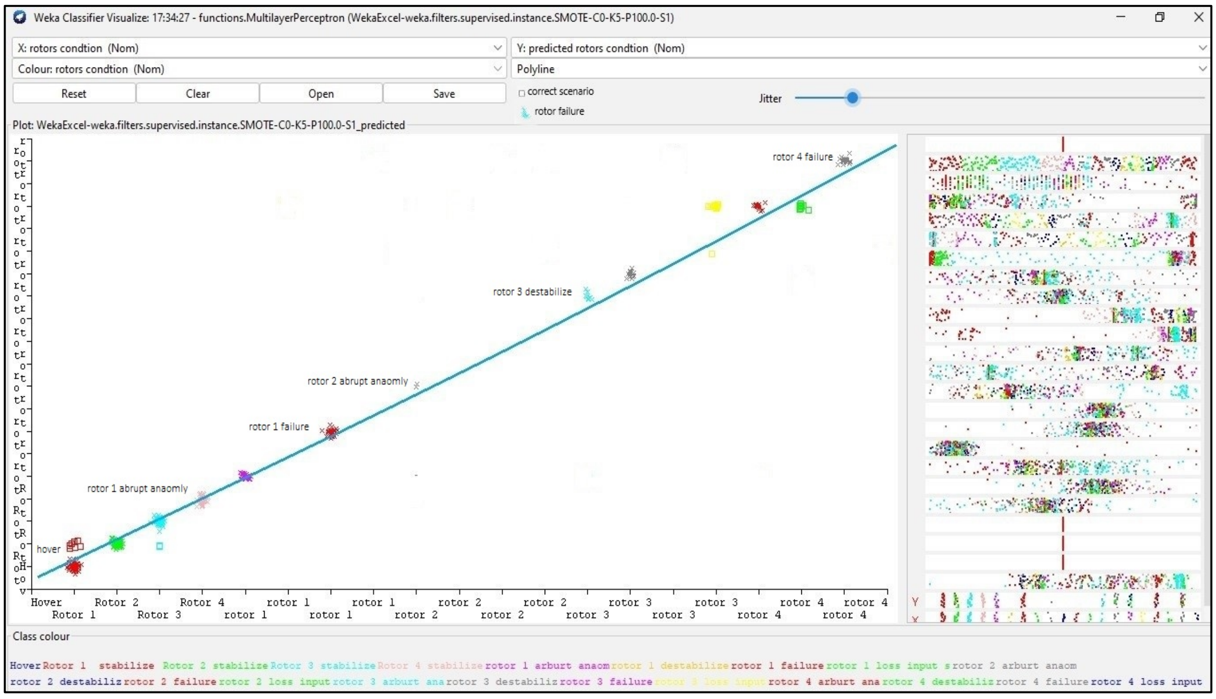
The CNN model implementation is shown in Fig. 10. The outcomes indicate that the design of the model and the chosen parameters were successful in increasing the classification accuracy for these categories on rotor failures.



Epoch 300
Num Of Epochs 300
Error per Epoch = 0.0188277
Learning Rate = 0.04
Momentum = 0.2

Figure 10. CNN Model Implementation

After implementing the model, the CNN classifier provides detailed data at each node of the rotor failure. The predicated and actual value curve of rotor failure shown in Figure 11.



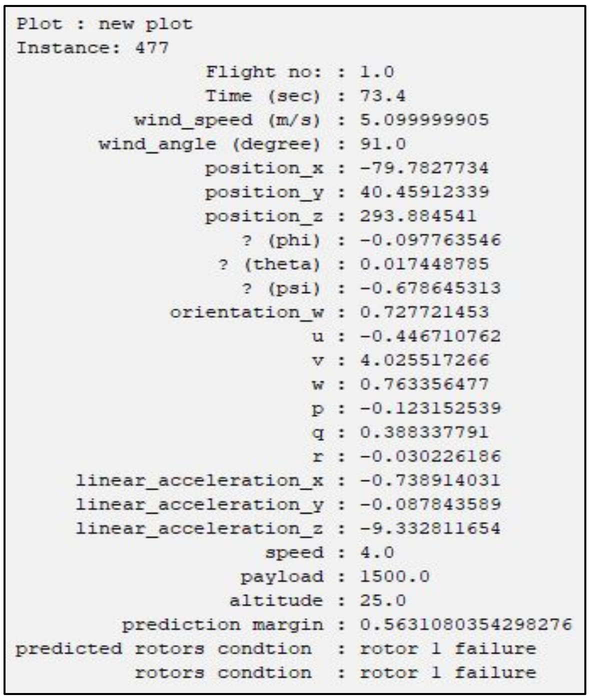
Weka Classifier Visualize: 17:34:27 - functions.MultilayerPerceptron (WekaExcel-weka.filters.supervised.instance.SMOTE-C0-K5-P100.0-S1)

X: rotors condtion (Nom)
Y: predicted rotors condtion (Nom)
Colour: rotors condtion (Nom)
Polyline
Jitter
Plot: WekaExcel-weka.filters.supervised.instance.SMOTE-C0-K5-P100.0-S1_predicted

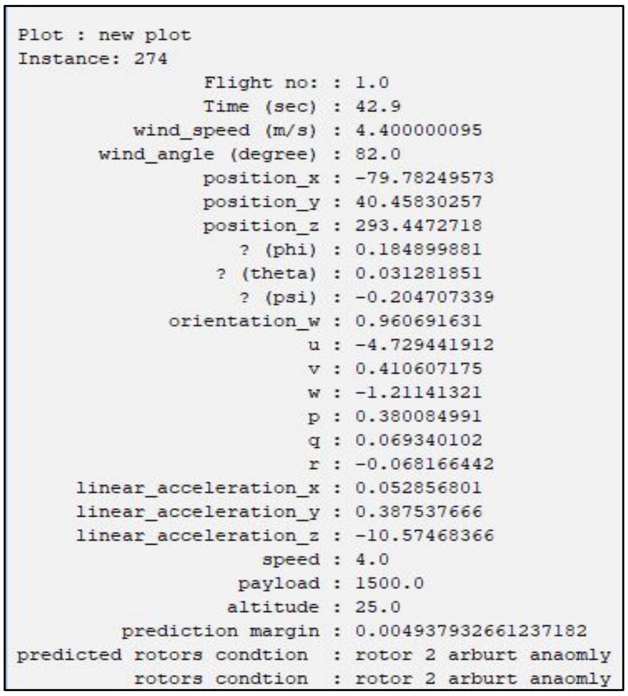
Reset
Clear
Open
Save
correct scenario
rotor failure
rotor 4 failure
rotor 3 destabilize
rotor 2 abrupt anaomly
rotor 1 failure
rotor 1 abrupt anaomly
hover
Hover
Rotor 1
Rotor 2
Rotor 3
Rotor 4
rotor 1
rotor 1
rotor 1
rotor 2
rotor 2
rotor 2
rotor 3
rotor 3
rotor 3
rotor 4
rotor 4
rotor 4
Class colour
Hover Rotor 1 stabilize Rotor 2 stabilize Rotor 3 stabilize Rotor 4 stabilize rotor 1 arburt anaom rotor 1 destabilize rotor 1 failure rotor 1 loss input srotor 2 arburt anaom
rotor 2 destabiliz rotor 2 failure rotor 2 loss input rotor 3 arburt ana rotor 3 destabiliz rotor 3 failure rotor 3 loss input rotor 4 arburt ana rotor 4 destabiliz rotor 4 failure rotor 4 loss input

Figure 11. CNN Model predicated and actual condition curve.

The different scenarios of rotor condition are visualized in Figure 11. The x-margin predicates the actual condition of rotor whereas, in y-margin the predicated condition is visualized.
At each point the CNN model provides detailed data of rotors as shown in fig 12 (a), (b), (c), (d).



```
Plot : new plot
Instance: 477
                 Flight no: : 1.0
                 Time (sec) : 73.4
          wind_speed (m/s) : 5.099999905
      wind_angle (degree) : 91.0
                 position_x : -79.7827734
                 position_y : 40.45912339
                 position_z : 293.884541
                    ? (phi) : -0.097763546
                  ? (theta) : 0.017448785
                    ? (psi) : -0.678645313
              orientation_w : 0.727721453
                          u : -0.446710762
                          v : 4.025517266
                          w : 0.763356477
                          p : -0.123152539
                          q : 0.388337791
                          r : -0.030226186
       linear_acceleration_x : -0.738914031
       linear_acceleration_y : -0.087843589
       linear_acceleration_z : -9.332811654
                      speed : 4.0
                    payload : 1500.0
                   altitude : 25.0
          prediction margin : 0.5631080354298276
predicted rotors condtion  : rotor 1 failure
          rotors condtion  : rotor 1 failure
```

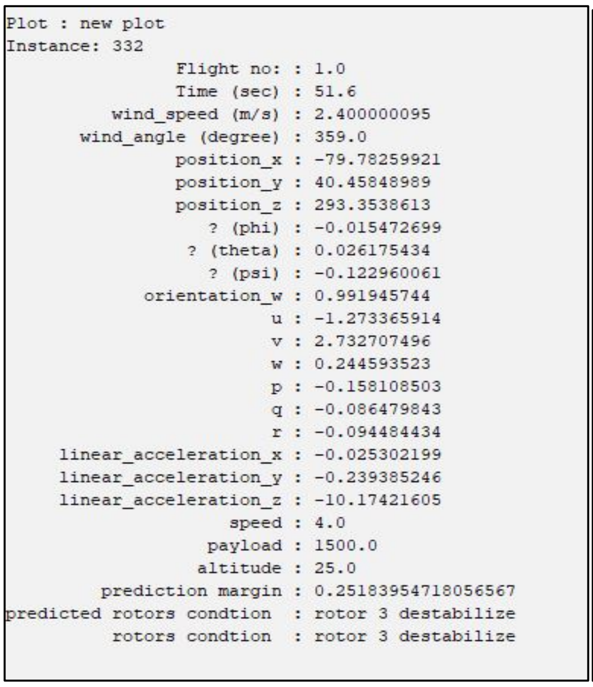


```
Plot : new plot
Instance: 274
                 Flight no: : 1.0
                 Time (sec) : 42.9
          wind_speed (m/s) : 4.400000095
      wind_angle (degree) : 82.0
                 position_x : -79.78249573
                 position_y : 40.45830257
                 position_z : 293.4472718
                    ? (phi) : 0.184899881
                  ? (theta) : 0.031281851
                    ? (psi) : -0.204707339
              orientation_w : 0.960691631
                          u : -4.729441912
                          v : 0.410607175
                          w : -1.21141321
                          p : 0.380084991
                          q : 0.069340102
                          r : -0.068166442
       linear_acceleration_x : 0.052856801
       linear_acceleration_y : 0.387537666
       linear_acceleration_z : -10.57468366
                      speed : 4.0
                    payload : 1500.0
                   altitude : 25.0
          prediction margin : 0.004937932661237182
predicted rotors condtion  : rotor 2 arburt anaomly
          rotors condtion  : rotor 2 arburt anaomly
```

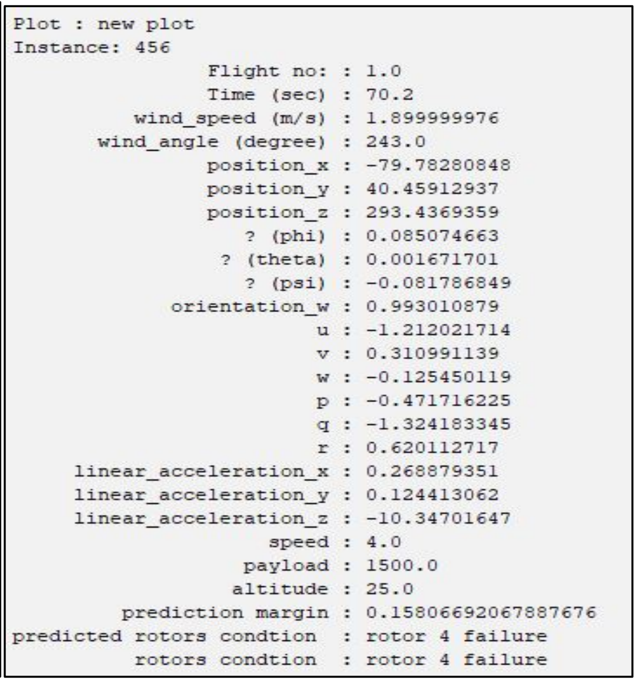
Figure 12 (a) Rotor 1

Figure 12 (b) Rotor 2

From figure 12 (a) we can conclude that our predicated value and actual value based on our given data are accurately classified. The rotor 1 failure occurs at given parameter with the mentioned parameters. Similarly for all the four rotors the same procedure is being carried out.



```
Plot : new plot
Instance: 332
                 Flight no: : 1.0
                 Time (sec) : 51.6
          wind_speed (m/s) : 2.400000095
      wind_angle (degree) : 359.0
                 position_x : -79.78259921
                 position_y : 40.45848989
                 position_z : 293.3538613
                    ? (phi) : -0.015472699
                  ? (theta) : 0.026175434
                    ? (psi) : -0.122960061
              orientation_w : 0.991945744
                          u : -1.273365914
                          v : 2.732707496
                          w : 0.244593523
                          p : -0.158108503
                          q : -0.086479843
                          r : -0.094484434
       linear_acceleration_x : -0.025302199
       linear_acceleration_y : -0.239385246
       linear_acceleration_z : -10.17421605
                      speed : 4.0
                    payload : 1500.0
                   altitude : 25.0
          prediction margin : 0.25183954718056567
predicted rotors condtion  : rotor 3 destabilize
          rotors condtion  : rotor 3 destabilize
```



```
Plot : new plot
Instance: 456
                 Flight no: : 1.0
                 Time (sec) : 70.2
          wind_speed (m/s) : 1.899999976
      wind_angle (degree) : 243.0
                 position_x : -79.78280848
                 position_y : 40.45912937
                 position_z : 293.4369359
                    ? (phi) : 0.085074663
                  ? (theta) : 0.001671701
                    ? (psi) : -0.081786849
              orientation_w : 0.993010879
                          u : -1.212021714
                          v : 0.310991139
                          w : -0.125450119
                          p : -0.471716225
                          q : -1.324183345
                          r : 0.620112717
       linear_acceleration_x : 0.268879351
       linear_acceleration_y : 0.124413062
       linear_acceleration_z : -10.34701647
                      speed : 4.0
                    payload : 1500.0
                   altitude : 25.0
          prediction margin : 0.15806692067887676
predicted rotors condtion  : rotor 4 failure
          rotors condtion  : rotor 4 failure
```

Figure 12 (c) rotor 3

Figure 12 (d) rotor 4

Figure 12. Detailed data of rotors failure (a) rotor 1 failure, (b) rotor 2 abrupt anomaly (c) rotor 3 destabilize (d) rotor 4 failure.

5. Conclusion & Future work:

This research developed Convolutional Neural Network model for detection of rotors failure based on the quadcopter IMU sensor data. Using MATLAB, a Simulink model of a quadcopter is taken to generate data under both normal and specified fault conditions by varying its accelerometer values. After collecting the data under safe and failure range by increasing the wind speed and its impact on quadcopter condition we generated training samples and their labels for our CNN algorithm. The data filtered under supervised learning. Later the data sample is divided for training and testing. The CNN based classifier is developed to classify rotors faults. All the rotor failures under CNN classifier show the exact values at which rotors of the quadcopter fail. The predicated conditions under the training data matched our testing sample. Our proposed model can classify various types of failure with 94% accuracy and is able to detect the condition of quadcopter along with fluctuated parameters and conditions.

As for future work, we intend to test our model in more crash/attack scenarios, such as those with propellers that are partially damaged but still work, cyberattacks, etc. Additionally, we intend to test our models on heterogeneous UAV platforms to evaluate their potential for generalizability. Later, we'll create methods for making decisions in real time to protect the drone from the identified error.

Funding: No external funding was received for this research.

Conflicts of Interest: The authors have no conflicts of interest to declare.

References

1. Abbaspour, A., Yen, K. K., Noei, S., & Sargolzaei, A. (2016). Detection of fault data injection attack on uav using adaptive neural network. Procedia computer science, 95, 193-200.
2. He, Y., Peng, Y., Wang, S., Liu, D., & Leong, P. H. (2017). A structured sparse subspace learning algorithm for anomaly detection in UAV flight data. IEEE Transactions on Instrumentation and Measurement, 67(1), 90-100.
3. Liu, Z., Yuan, C., Yu, X., & Zhang, Y. (2019). Retrofit fault-tolerant tracking control design of an unmanned quadrotor helicopter considering actuator dynamics. International Journal of Robust and Nonlinear Control, 29(16), 5293-5313.
4. Stutz, J. (2010). On data-centric diagnosis of aircraft systems. IEEE Trans Syst Man Cybern–PART C.
5. Sadhu, V., Salles-Loustau, G., Pompili, D., Zonouz, S., & Sritapan, V. (2017, March). Argus: Smartphone-enabled human cooperation for disaster situational awareness via MARL. In 2017 IEEE International Conference on Pervasive Computing and Communications Workshops (PerCom Workshops) (pp. 79-81). IEEE.
6. Deng, D., & Yuan, H. (2015, November). UAV flight safety ground test and evaluation. In 2015 IEEE AUTOTESTCON (pp. 422-427). IEEE.
7. Yu, B., Zhang, Y., & Qu, Y. (2015, October). MPC-based FTC with FDD against actuator faults of UAVs. In 2015 15th International Conference on Control, Automation and Systems (ICCAS) (pp. 225-230). IEEE.
8. He, Z., Wei, J., & Hou, B. (2018, May). Detecting Incipient Faults in Quad-rotor Unmanned Aerial Vehicle Based on Detrending and Denoising Techniques. In 2018 IEEE 7th Data Driven Control and Learning Systems Conference (DDCLS) (pp. 959-964). IEEE.
9. Wang, B., Liu, D., Peng, Y., & Peng, X. (2019). Multivariate regression-based fault detection and recovery of UAV flight data. IEEE Transactions on Instrumentation and Measurement, 69(6), 3527-3537.
10. Zhao, B., Cheng, C., Peng, Z., Dong, X., & Meng, G. (2020). Detecting the early damages in structures with nonlinear output frequency response functions and the CNN-LSTM model. IEEE Transactions on Instrumentation and Measurement, 69(12), 9557-9567.
11. Ma, Y., & Li, H. (2020). GRU-Auto-Encoder neural network based methods for diagnosing abnormal operating conditions of steam drums in coal gasification plants. Computers & Chemical Engineering, 143, 107097.
12. Hu, J., Kuang, Y., Liao, B., Cao, L., Dong, S., & Li, P. (2019). A multichannel 2D convolutional neural network model for task-evoked fMRI data classification. Computational intelligence and neuroscience, 2019.
13. Liu, Z., Cao, J., Lu, Y., Chen, S., & Liu, J. (2019). A seismic facies classification method based on the convolutional neural network and the probabilistic framework for seismic attributes and spatial classification. Interpretation, 7(3), SE225-SE236.
14. Dokur, Z., & Ölmez, T. (2020). Heartbeat classification by using a convolutional neural network trained with Walsh functions. Neural Computing and Applications, 32, 12515-12534.
15. Pairan, M. F., Shamsudin, S. S., Yaakub, M. F., & Anwar, M. S. M. (2021). Real-time system identification of an unmanned quadcopter system using fully tuned radial basis function neural networks. International Journal of Modelling, Identification and Control, 37(2), 128-139. DOI.org (Crossref), https://doi.org/10.1504/IJMIC.2021.120209.
16. Padhy, R. P., Verma, S., Ahmad, S., Choudhury, S. K., & Sa, P. K. (2018). Deep neural network for autonomous uav navigation in indoor corridor environments. Procedia computer science, 133, 643-650. , https://doi.org/10.1016/j.procs.2018.07.099.
17. Debele, Y., Shi, H. Y., Wondosen, A., Ku, T. W., & Kang, B. S. (2023). Deep Learning-Based Robust Actuator Fault Detection and Isolation Scheme for Highly Redundant Multirotor UAVs. Drones, 7(7), 437. https://doi.org/10.3390/drones7070437.