Real-Time Digital Twin Implementation for Predictive Maintenance of UAVs Using Embedded Sensor Fusion and Machine Learning

Naren Pai[1]

William G. Enloe High School, Raleigh, North Carolina, 27610, USA

Small unmanned aerial vehicles (UAVs) operate under sustained vibration and cyclic loading, leading to gradual degradation of propulsion components. Existing maintenance strategies are largely reactive and do not provide real-time insight into subsystem health. This work presents a subsystem-level digital twin architecture that uses onboard sensor fusion and machine learning to perform real-time fault classification for UAV propulsion systems. Multivariate time-series data were collected from nine onboard sensor channels, including tri-axis acceleration, motor current, shaft angle, and commanded PWM. Data were recorded in 60 s experimental runs across four different issue categories representing nominal operation, induced mass imbalance, and propeller damage, with a severity value encoded from 0 to 3. A Long Short-Term Memory (LSTM) network extracts temporal features from fixed-length sensor windows (L = 64 samples), and the embeddings are classified using Random Forest models for both issue and severity prediction. Model evaluation demonstrates recall of 1.00 for propeller damage and 0.98 for imbalance faults, while nominal conditions display partial overlap. Severity classification achieved near-perfect accuracy for levels 0 and 3. These results demonstrate that lightweight embedded models using only onboard sensors can provide effective real-time propulsion fault detection, establishing the feasibility of subsystem-level digital twins for predictive maintenance in small UAV platforms.

I.Introduction

Small unmanned aerial vehicles are increasingly deployed for missions that require repeated flights under varying mechanical loads. During operation, components such as motors, bearings, and control linkages experience gradual degradation caused by vibration, thermal stress, and repeated actuation. These changes often remain undetected by flight operators until performance degrades sharply or a failure occurs during flight. As a result, many Unmanned Aerial Vehicle (UAV) failures provide little or no advance warning to operators.

Maintenance practices for small UAVs typically rely on fixed inspection schedules or post-failure diagnosis. While this approach is adequate for low frequency operation, it does not provide real time insight into vehicle health during flight and does not scale to larger deployments. Studies of rotating machinery have shown that faults such as imbalance and bearing wear can evolve over time while remaining within nominal control limits until a critical threshold is reached [1]. As a result, periodic inspection cannot capture the evolving faults that emerge under operational loads, particularly in small UAVs that experience high vibration and rapid duty cycles. This maintenance gap motivates the need for continuous health monitoring.

II.Related Work

Maintenance and fault detection for aerial systems has traditionally followed two primary approaches: schedule-based inspection and external inspection using visual sensing. Schedule based maintenance strategies aim to reduce failure probability by increasing inspection frequency or optimizing inspection intervals based on operational constraints. For example, Petritoli, Leccese, and Ciani [2] present a computational framework to optimize UAV

[1] Student, High School AIAA Member, ID: 1857147

maintenance scheduling based on fleet size and personnel availability. While such approaches improve logistical efficiency, they remain inherently reactive and do not provide insight into vehicle health during flight.

A second class of approaches focuses on external inspection using computer vision systems to identify structural defects. Vision based inspection has demonstrated effectiveness in detecting surface level damage on surfaces where defects such as cracks are visually observable [3]. However, small UAV platforms exhibit significant variation in geometry, materials, and layout, limiting the generalization of a single vision model across platforms. More importantly, visual inspection cannot observe behaviors within components, which are common precursors to failure in small UAVs.

In contrast, extensive research in rotating machinery diagnostics has shown that early-stage faults in motors and bearings produce measurable changes in vibration spectra and current draw well before catastrophic failure occurs [1,4]. These methods rely on time series analysis of sensor data to detect deviations from nominal behavior. Similar principles have been applied to anomaly detection using recurrent neural networks, particularly long short-term memory models, which are well suited for modeling temporal dependencies in sensor signals [5]. Despite these advances, the application of time series-based fault detection to small UAVs remains largely unexplored. This motivates a digital twin approach that models UAV subsystems independently using only onboard sensors, enabling early fault detection while remaining adaptable across platforms

III.System Overview

A. Digital Twin Concept

This project proposes a subsystem-level digital twin for small unmanned aerial vehicles that continuously models normal operational behavior using onboard sensor data. In this work, digital twin is defined as a data-driven model that learns the expected temporal behavior of a physical subsystem and flags deviations from that behavior during operation. Rather than attempting to replicate the full UAV, the system models individual subsystems independently, allowing the approach to generalize across different airframe configurations.

The digital twin works on two common UAV subsystems: the propulsion subsystem and the control actuation subsystem. While this work experimentally evaluates the performance of a digital twin for the propulsion subsystem, the proposed architecture is designed to generalize to additional UAV subsystems such as control actuators and linkages. For the propulsion subsystem, time series data are collected from inertial measurement units, motor encoders, and current sensors mounted near the motor. These sensors are commonly available on small UAV platforms and easy to integrate. For each subsystem, a long short-term memory neural network is trained to model nominal behavior using time series sensor data collected during normal operation. During operation, incoming sensor data is passed through the trained model to classify an error. Deviations beyond a defined threshold are flagged as abnormal behavior. To evaluate the system, controlled faults are introduced into each subsystem. [1, 4]

B. Propulsion Subsystem

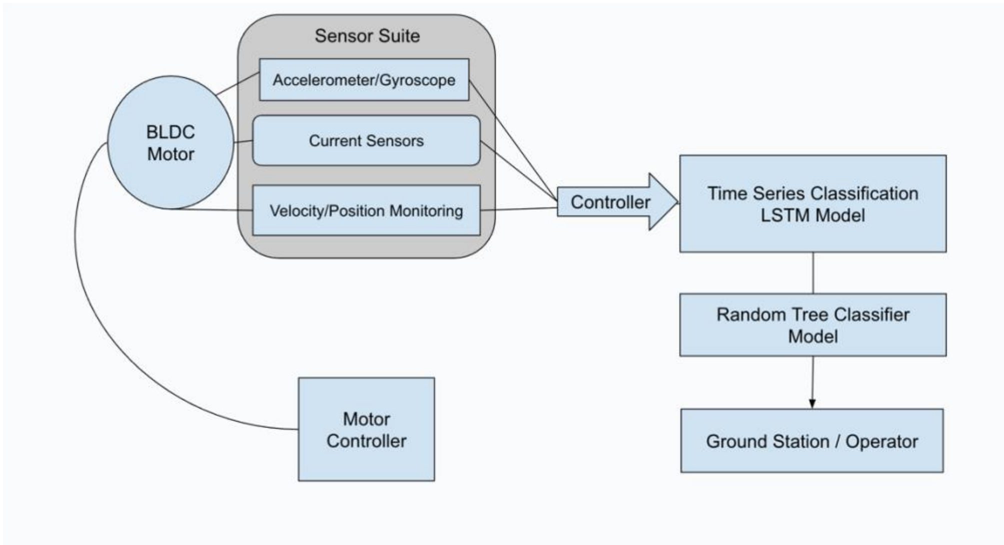

Fig.1 A depiction of the sensor suite and analysis for the propulsion subsystem

The parts selection was carried out by selecting parts typically used in 9-10 in quadcopters.

Table 1. Propulsion subsystem components and specifications.

Component	Purpose	Key Specs
MPU6050	Vibration and angular motion	±16 g accel; ±2000 dps gyro
ACS724	Motor current	0–24 V; bidirectional
MT6701	Shaft angle	14-bit absolute encoder
A2212 motor	Test motor	1000 KV
Electronic Speed Controller (ESC)	Motor drive	30 A

C. Mechanical Layout
A 3D printed frame was used to mount all respective electronic. The MPU6050 Inertial Measurement Unit (IMU) was positioned 35 mm away from the motors drive shaft. On the bottom of the motor shaft, a diametrically magnetized magnet was mounted using epoxy. Within 1 mm of the magnet, the MT6701 encoder was mounted.

D. Data Acquisition Hardware
An Arduino Uno microcontroller board was used to interface with the sensors. The IMU and encoder communicated data with the I2C protocol. The current sensor provided an analog output with an output range from 0 to 1,023. The ESC speed value was set with Pulse Width Modulation (PWM). To log the data of every run, a micro-SD card was interfaced with the Arduino with the SPI protocol to collect data in the comma-separated values (CSV) format.

IV.Data Collection

A. Operating Conditions
For each experiment, data was set to be logged for a timespan of 60 seconds, with the ESC set to a constant PWM value. The subsystem was clamped down to a table with a constant pressure, in constant temperature room.

B. Fault Types

Data was collected on a set of propulsion fault categories, which represent damage mechanisms. Offset mass faults were added to the system through epoxy mounted masses. Faults were defined and paired with a severity value. The severity of each fault category is encoded using a value from 0 to 3. The value indicates the increasing physical deviation from normal operation

Table 2. Propulsion subsystem fault and severity level

Fault	Severity Level (0-3)
Nominal (No propeller)	0
Mass imbalance, .3 grams	1
Mass imbalance .6 grams	2
Mass imbalance 1 gram	3
Nominal (With propeller)	0
Propeller Damage	3

C. Dataset

For each logged run, a CSV file was collected of timestamped data in milliseconds, for a 60 second run. 3 runs were collected for each fault condition. For each sample, the following information was collected

Table 3. Sensor Data label and value

Data Label	Value
timestamp_ms	Row timestamp in milliseconds
ax	X-axis accelerometer value
ay	y-axis accelerometer value
az	z-axis accelerometer value
gx	X-axis gyroscope value
gy	y-axis gyroscope value
gz	z-axis gyroscope value
curr_a	Motor Current draw

Abs_angle	Absolute encoder angle
Motor_pwm	ESC PWM value
issue	Issue name
severity	Issue severity
run_id	Run number

V.Model Architecture

A. LSTM Network

The propulsion subsystem deploys a Long Short-Term memory (LSTM) recurrent neural network to model the deviations in multivariate sensor signals. LSTM networks are particularly suited for these applications due to their ability to capture the long-range relationships and developing patterns in timeseries data, which are characteristic of the vibration, current, and rotational signals produced by rotating motors. [6]

Each input sample to the network has a fixed-length time window of sensor measurements

$$\boldsymbol{X} \in \mathbb{R}^{L \times F}$$

Where $\mathbb{R}$ is the set of real numbers, L is the window length and $F = 9$ is the number of sensor channels. The default window length used in this study is $L = 64$ samples, with a stride of 16 samples between windows. The LSTM architecture consisted of one network layer with 64 units, and an output embedding vector of dimension 64.

B. Random Forest Classifier

Two, Random Forest classifiers operate on the LSTM model outputs: an issue classifier and a severity classifier. Each classifier receives a vector of length 64 as the input, and outputs a discrete class label. 500 decision trees are used with majority voting. The final prediction for each window is taken as the class with maximum predicted probability. A confidence threshold is applied, with those whose probabilities falling below 0.5 marked as unknown, rather than an inaccurate classification.

C. Training Procedure

All sensor channels were standardized using a z-score normalization. Data was grouped with overlapping windows of L = 64, and a stride of 16. Only the sensor data columns were used as the model inputs, and none of the issue and severity labels were included in the input.

1. *LSTM Training*

   Training was carried out with the sparse categorical cross-entropy loss function. The batch size was 128, and training was conducted for 20 epochs.

2. *Random Forest Training*

   After the LSTM was trained, embeddings were extracted for all windows and trained the two issue and severity classifiers.

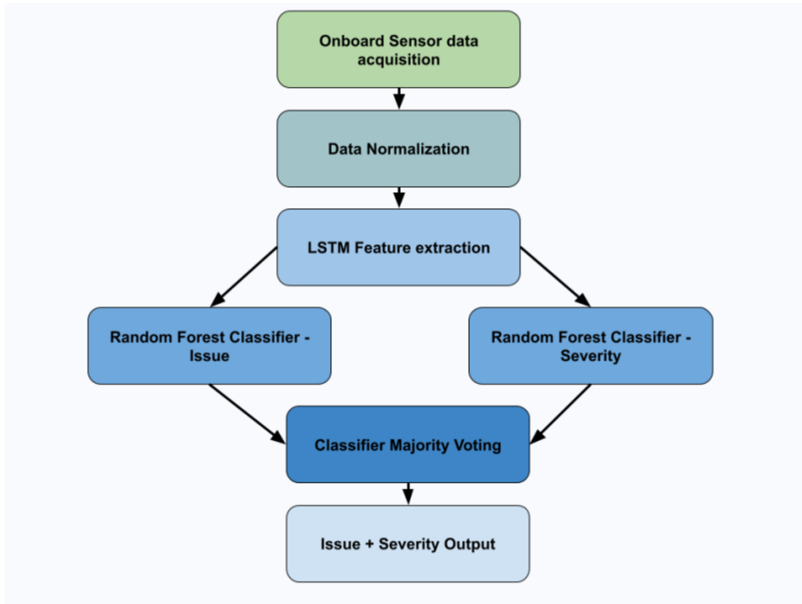

Fig.1 Model issue classification flow

VI. Experimental Results

Experiments were conducted using time-series data collected from the propulsion test platform under multiple operating conditions and fault types.

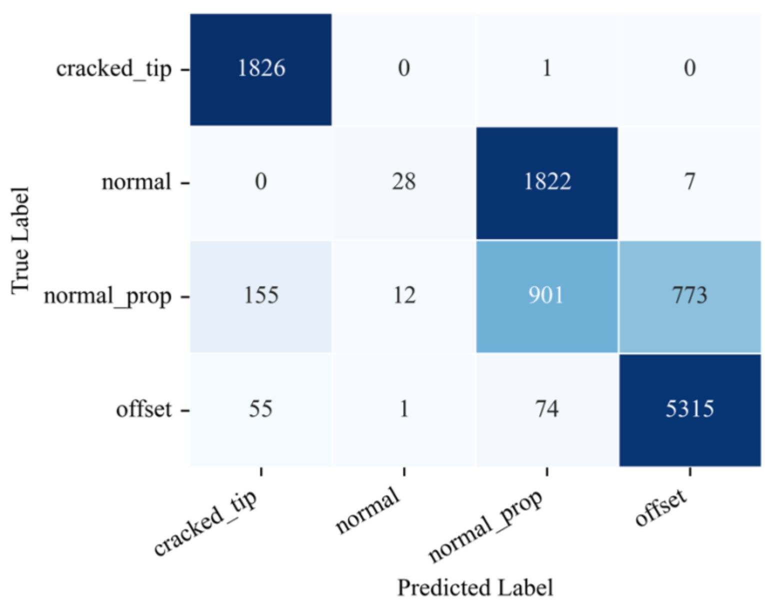

Fig.2 Confusion Matrix for Issue Classification

Figure 3 displays confusion matrix for issue classification. The model achieves strong performance for the cracked tip and offset mass conditions, with 1826 and 5315 correctly classified windows. The model misclassifies between the normal operation and normal operation with propeller conditions, indicating that with-prop conditions were not clearly learned. The results indicate that the learned model effectively could separate fault conditions from normal operation but fails to precisely separate overlap between normal operating conditions.

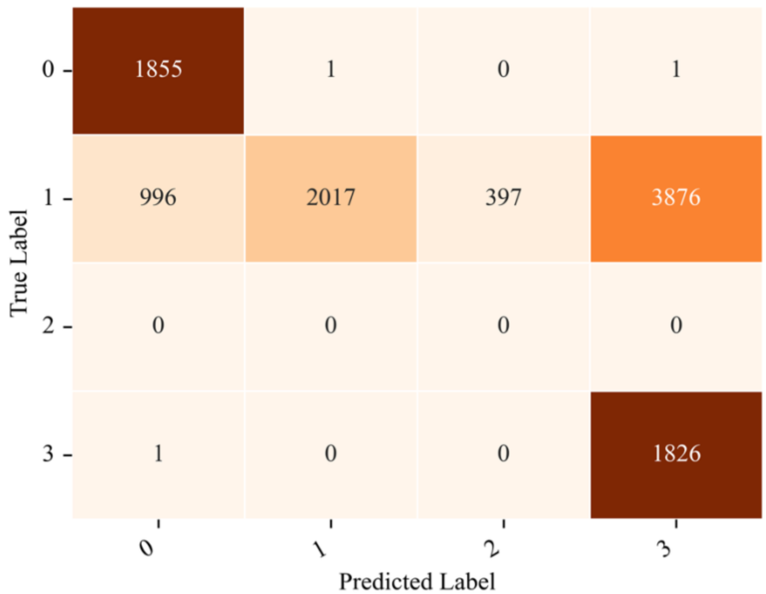

Fig.3 Confusion Matrix for Severity Classification

Figure 4 presents the severity classification confusion matrix. The model was able to accurately identify nominal (0) and high-severity (3) conditions with almost perfect accuracy, however moderate severity (1) displays confusion with nominal and severe classes. This suggests that transitions occur on a spectrum, rather than discrete boundaries, which may make the intermediate severities more difficult to classify.

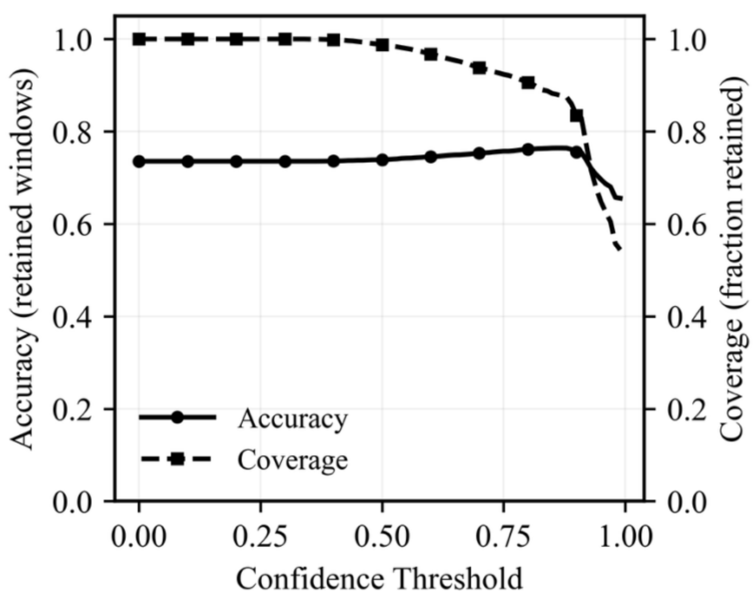

Fig.5 Accuracy vs. Coverage Trade-off

Figure 5 displays the trade-off between accuracy and coverage, as the confidence threshold is varied. Increasing the threshold reduces coverage, while slightly increasing accuracy, up to a threshold of approximately 0.85. This demonstrates the feasibility of using thresholds to control the model's reliability in deployment.

VII.Discussion

A. Fault Discrimination

The issue classification results show the high discrimination of the model for propulsion faults, specifically with the cracked propeller tip and the offset mass. The classes exhibit a high recall, indicating that the LSTM learned the temporal relationships associated with the damage. These signatures are associated with the vibration amplitude and current draw. The offset mass introduces periodic amplitude modulation, while the cracked propeller introduces asymmetric aerodynamic loading.

However, confusion between the normal operating conditions, and normal operating conditions with the propeller, indicate overlapping vibration patterns between operating conditions. Future work is required for the model to be able to contextualize other features, to isolate the addition of a propeller.

B. Severity Modeling

The severity classification displays a near-perfect performance for nominal and high-severity conditions. However, moderate severity displays confusion with adjacent values. This displays the evolution of severity over the period of motor operation. The intermediate severity levels likely lie on a continuous spectrum, rather than discrete clusters, reinforcing the need for interpretation of severity as a variable which progresses, rather than a single category.

C. Limitations

Testing was conducted on a ground-based propulsion platform rather than in in-flight conditions. Introduction of aerodynamic disturbances and structural interference will introduce additional variability. In addition, a limited number of fault types were evaluated. Broader validation is needed for the array of possible issues in all components of the system. Further, the limited number of experimental runs per fault category restrict generalizability. Additionally, limits in sensor capabilities for sampling frequency may have also limited the model's sensitivity to high-frequency vibrations.

D. Implications for UAV Health monitoring

Despite the limitations, the results of the model testing support the premise of this study: the development of a sub-system level digital twin utilizing the onboard sensor suite to detect propulsion faults in real time. This architecture enables the direct classification of known fault modes and respective severity levels, suitable for embedded deployment on common flight hardware. The temporal monitoring of the system and issue and severity classification provide a practical framework for a potentially scalable framework for comprehensive UAV health modeling. This approach can be extended to additional subsystems, which enables a modular approach to UAV maintenance, while collectively improving vehicle reliability.

VIII.Future Work

A. Expanded Dataset and Operation Diversity

Future work will focus on expanding the dataset across a broader array of operating conditions, and longer time horizons. The current study only evaluates a controlled set of propulsion faults, which are under stable ground conditions. Widening the dataset would improve the model's robustness for the large set of conditions. Additionally, incorporating data from multiple motors of the same type across different manufacturing batches can allow for an evaluation of the transferability for models and reduce overfitting for a single hardware issue.

B. In-Flight Validation

The present evaluation was conducted on a ground-based propulsion platform, which allows for controlled experimentation. This however does not take into consideration the multitude of factors introduced with inflight conditions. Future work will integrate the digital twin onto a quadcopter platform to evaluate its viability and performance during real flight missions.

C. Continuous Severity Estimation

In this experiment, severity levels were discretely divided into 4 classes. Future work will explore continuous severity modeling through regression-based modeling. Evaluating fault progression as a continuous variable will allow for a better reflection of the physical degradation process occurring over time in the system.

References

[1] Randall, R. B., *Vibration-Based Condition Monitoring: Industrial, Aerospace and Automotive Applications*, John Wiley & Sons, Chichester, UK, 2011.

[2] Petritoli, E., Leccese, F., and Ciani, L., "Reliability and Maintenance Analysis of Unmanned Aerial Vehicles" *Sensors*, Vol. 18, No. 9, 2018, Article 3171.

[3] Malche, T., Tharewal, S., and Dhanaraj, R. K., "Automated Damage Detection on Concrete Structures Using Computer Vision and Drone Imagery," Engineering Proceedings, Vol. 58, No. 1, 2023, Article 60.

[4] Thomson, W. T., and Fenger, M., "Current Signature Analysis to Detect Induction Motor Faults" *IEEE Industry Applications Magazine*, Vol. 7, No. 4, July–Aug. 2001, pp. 26–34.

[5] Malhotra, P., Vig, L., Shroff, G., and Agarwal, P., "Long Short-Term Memory Networks for Anomaly Detection in Time Series" *Proceedings of the 23rd European Symposium on Artificial Neural Networks, Computational Intelligence and Machine Learning (ESANN)*, 2015.

[6] Staudemeyer, R. C., and Morris, E. R., "Understanding LSTM—A Tutorial into Long Short-Term Memory Recurrent Neural Networks" *arXiv preprint*, arXiv:1909.09586, 2019.

[7] Wild, G., Murray, J., and Baxter, G., "Exploring Civil Drone Accidents and Incidents to Help Prevent Potential Air Disasters" *Aerospace*, Vol. 3, No. 3, 2016, Article 22.

[8] Arduino IDE, Integrated Development Environment, Ver. 2.2.1, Arduino SA, Turin, Italy, 2023.