Unsupervised Anomaly Detection Using Batteries in Electric Aerial Vehicle Propulsion Test-Bed

John Pace[1], Jubilee Prasad Rao[2], Jesse Williams[3], and Liang He[4]

[1,2,3] *Global Technology Connection, Inc, Atlanta, GA, 30067, USA*
jpace@globaltechinc.com
jrao@globaltechinc.com
jwilliams@globaltechinc.com

[1,4] *University of Colorado Denver, Colorado, 80204, USA*
liang.he@ucdenver.edu

Abstract

There is a growing variety of manned and unmanned aerial vehicles that utilize batteries as their primary power source. These vehicles are composed of a large variety of interacting components and sensors that are needed for safe operation and to carry out their respective missions. As their interactions, complexity, and numbers increase, the risk for anomalies such as degradation of components, sensor faults, and erroneous controls also increase. These anomalies pose significant risks for vehicles flying over densely populated areas or conducting critical missions. It is therefore crucial to detect and mitigate these anomalies. There exist several approaches for anomaly detection, such as traditional rule or threshold-based methods, model-based approaches, supervised machine learning-based methods, and even unsupervised methods to detect different types of abnormal behaviors. These methods have inherent drawbacks including lack of sensitivity, inability to detect previously unknown faults, not being robust to compromised in-network information, or requiring sophisticated system models. To this end, we propose BDAV, a Battery-based Diagnosis for Aerial Vehicles, that uses machine learning models to learn physics-based dependencies and an unsupervised algorithm to detect and identify anomalies. BDAV is inspired by the physical dependencies between a vehicle's operation and the concomitant power consumption, allowing the use of battery as a trustworthy sensor to detect anomalies in a vehicle's operation(a hardware-based root-of-trust). Specifically, BDAV utilizes features extracted from run-time battery voltage and current information to construct learning models (norm maps) that map dependencies independently between battery metrics and each system operational variable. During operations, these norm maps allow operational variable values to be estimated assuming non-anomalous behavior. Some residual errors are expected in these predictions, however the cumulative sum of the error between the predictions and observed values is expected to follow a linear trend during normal operation of a system. Anomalies are detected when deviations in this trend are observed, which are quantified using five key parameters of the unsupervised anomaly detection algorithm. Preliminary optimization of BDAV parameters and testing on an electric-propulsion testbed demonstrated anomaly detection rates up to 91% and false positive rates as low as 2.5% for operational variables such as propeller thrust, motor rpm, and system vibration across a variety of injected anomaly types. This approach is applicable to most systems with electric batteries and can be rapidly optimized and adapted for efficient and cost-effective onboard fault management.

1. Introduction

Electric aerial vehicles are a rapidly growing field with an ever increasing number of uses ranging from military applications, spatial mapping (Bemis et al., 2014), agriculture (Velusamy et al., 2021), package delivery (Thiels, Aho, Zietlow, & Jenkins, 2015), Urban Air Mobility (UAM) (Silva, Johnson, Solis, Patterson, & Antcliff, 2018), or even the construction of temporary mobile networks (Moradi, Sundaresan, Chai, Rangarajan, & Mao, 2018; Chakraborty, Chai, Sundaresan, Khojastepour, & Rangarajan, 2018). These applications demand different vehicular designs and payload capabilities which include actuators, sensors, onboard computers, and other components (Swartz, 2017). In sophisticated applications, such vehicles are fairly complex and comprise of hundreds of different components. There also exist several types of flight controllers (Ebeid, Skriver, & Jin, 2017) for perception, vehicle control, and communication functions in aerial vehicles. They use data from vehicle sensors, onboard components, and sometimes also a human controller to adjust motor speeds to control the vehicle. Despite some built-in redundancies, all the components are required to operate as designed to achieve required performance and avoid safety incidents (Quinones-Grueiro, Biswas, Ahmed, Darrah, & Kulkarni, 2021). Failure of components, faulty sensors, inclement weather, or even errors within the flight control software could result in erratic behavior, mismanagement of systems, or even complete failure and crashes (Quinones-Grueiro et al., 2021; Pesé, Ganesan, & Shin, 2017; Wasicek, Pese, Weimerskirch, Burakova, & Singh, 2017; Bai, ElBatt, Holland, Krishnan, & Sadekar, 2006; ElBatt, Goel, Holland, Krishnan, & Parikh, 2006; Jones, 2002; Waraksa, Fraley, Kiefer, Douglas, & Gilbert, 1988; Diem, 2001; Feser, McConnell, Brandmeier, & Lauerer, 2006). These may lead to damage to the vehicle and nearby property as well as harm to humans in the current or nearby vehicles or on the ground (Bauranov & Rakas, 2019). Therefore it is critical to detect and identify any sources of anomalies on the vehicle, determine mitigating strategies, and implement them in a timely manner.

Traditional methods to detect anomalies on systems rely on thresholds to measured sensor values or derived operational variables. However, there exist circumstances in which sensor readings are within expected ranges but the behavior demonstrated by that sensor is anomalous, leading to false negatives. This results in a delayed detection or unexpected eventual failure of a component and its corresponding consequences. Anomaly detection methods based on learning models rely on in-vehicle data to predict the behavior of other in-vehicle parameters, or are trained to recognize specific abnormal patterns. These approaches have multiple drawbacks. First, there exist situations where the entire system itself could be compromised, either due to a cyberattack or complete system error. In such scenarios, the data sources that these models rely on to diagnose anomalies may become unreliable, i.e., the diagnostic systems themselves could be abnormal (Miller & Valasek, 2015; Cho & Shin, 2016; Lanigan, Kavulya, Narasimhan, Fuhrman, & Salman, 2011). Secondly, learning models trained to recognize specific known anomalies must have first been exposed to that anomalous behavior, rendering them incapable of diagnosing new types of anomalies the model has not yet seen (Choi et al., 2016; Murvay & Groza, 2014; Baker, Ferguson, & Dolan, 2008).

To address these issues, we design BDAV, a diagnostic system for battery powered aerial vehicles that utilizes vehicle's battery as root of trust. BDAV is built on the fact that subsystems and measured sensor values of an aerial vehicle have physically-induced dependencies, observable at the battery, that persist throughout the vehicle's lifetime. By capturing these dependencies, predictions about expected vehicle operation can be made directly and using only battery information. The expected values are then compared to measured system behavior and deviations quantified and used to detect the presence of anomalies.

Electric batteries have unique advantages that make them promising root of trust. They are almost ubiquitous in most aerial vehicles (He, Kong, Liu, Shu, & Liu, 2019), and their voltage and current data can be measured directly and reliably from the physical component. They can be measured without modifying the vehicle's internal systems or hardware design using inexpensive sensors, which reduces the cost of implementation. Batteries themselves can also become anomalous, and several battery diagnosis algorithms exist (Tran & Fowler, 2020) to detect battery faults. This work assumes that a vehicle's battery is operating nominally and uses it as ground truth. The diagnostic sensor also needs to have a large coverage and have inter-dependencies with as many system components as possible. In most vehicle designs, one or a set of interconnected batteries provide power for all vehicular functions. Hence, battery current has a strong relationship with most system components, making it suitable to estimate vehicle state and validate system operational variables. Measuring this root-of-trust battery information in physical isolation of the internal network is needed (He et al., 2019) to add a layer of separation and increase confidence in diagnosis during cyberattacks on the in-vehicle network itself. BDAV is a widely applicable solution and could be deployed on virtually all electrically powered aerial, ground, underwater, and even space vehicles. It could also be deployed to only a subsystem or component of a system, and does not require excessive system reconfiguration or tuning. In this paper, we demonstrate the feasibility of this approach by implementing it on an electric propulsion testbed comprising of one Brush-Less Direct Current (BLDC) motor connected to a two bladed propeller, an Electronic Speed Controller (ESC), powered by a Lithium-Ion battery, and manually controlled through a servo controller. The testbed has several sensors to measure battery current and voltage, thrust, motor RPM and vibration generated by the system.

To describe the BDAV approach and demonstrate its application to the testbed, this paper is organized as follows. Section 2 describes feature extraction from battery voltage and current data and training of norm models to predict other sensor data. Section 3 develops the error handling methodology and a framework to detect anomalies based on error accumulation and its slope. Section 4 presents the electrical propulsion test bed, application of the developed anomaly detection methodology to it, and its results. Section 5 presents a conclusion to the conducted work and a direction for future work.

2. PREDICTING SYSTEM NORMAL BEHAVIOR

To detect anomalies that present themselves as aberrations from normal behavior, using an unsupervised approach, a system's normal behavior must be learnt. During operations this expected behavior is predicted and compared to a system's observed behavior, which can give insight into the presence of any anomalies. To achieve this, for any application, a nominal amount of system operational data without any anomalies is needed. The amount of data should be sufficient to cover all general states to be encountered by the system during operation. Lack of sufficient data may result in false positives when the system undergoes a state or series of states significantly different from those in the training data sufficient to be identified as an anomaly by the algorithm. Part of the available data is to be set aside for testing the developed algorithms. In this section, we develop a method to extract features from battery voltage and current data.

2.1. Battery Feature Extraction

Machine learning models learn complex relationships between different available features and a target variable using a large number of instances to estimate target values for new sets of features (Mitchell & Mitchell, 1997). For BDAV, features are to be extracted from the battery voltage and current, and are needed to train ML models to predict system operational variable like motor RPM, thrust, and others.

2.1.1. Time Window Construction

In systems like electric aerial vehicles, current draw from the battery is very dynamic, and current and voltage readings from one time instance are not sufficient to identify the state of the vehicle and much less to predict another system operational variable. Instead, a time window that examines battery data preceding the most recent reading may be more useful. A moving time window is used to characterize the last *n* seconds of battery information. When a new battery reading is generated, a time window is constructed over $[T_{latest} - T_w, T_{latest}]$, with T_w being the size of the window. For each new reading the window is updated with the newest battery samples, and samples no longer within the time period are removed. Within this time window we characterize several features of battery current and voltage, including arrays of all local minimums and maximums (referred to as craters and peaks) within current, averages, and absolute minimums and maximums along with their respective timestamps are extracted. When determining the size of the window examined, there is a trade-off between run-time performance and amount of historical characterizing information. This depends on the data acquisition rate and the system dynamics.

2.1.2. Peak Detection

Within current and voltage measurements, there exist small fluctuations due to sensor noise and the granularity given by the analog range of a micro-controller (i.e., 0–1023). These fluctuations generate false peaks and craters within the trace. To remove this noise, a low pass filter is needed as a data preprocessing step. After filtering, current trace is checked in for peaks and craters. By filtering the data first, we can monitor the trend in current as either increasing or decreasing, and a peak or crater is identified when the trend direction changes. Peaks and craters are characterized by their amplitudes and timestamps, making each a tuple of $\{a, t\}$.

2.2. Norm Model Construction

A machine learning approach to training norm models is selected to rapidly generate one-one maps from battery features to all necessary system operational variables. Each of the machine learning model will correspond to one operational variable and predict what values are expected for it under normal operational conditions. To train these machine learning models, a feature set constructed from each updated time window characterizing battery voltage and current data is utilized. That is, each new current and voltage reading collected by the external micro-controller results in a feature vector, $F = \{f_1, \ldots f_7\}$, characterizing battery information over the last $T_{now} - T_w$ readings. f_1 and f_2 contain the most recent current and voltage readings taken at time T_{now} for the given window. f_3 and f_4 hold the most recent peak and crater tuples, found as the last element added to the lists of peaks and craters over the time window. f_5 and f_6 cover the absolute minimum and maximum amplitudes during the time period, while f_7 is the mean of the current readings in the present time window. A target value, g_i, is then added for each feature vector and synchronized by the timestamp. It corresponds to the operational variable under consideration and could be battery temperature, vehicle acceleration, motor RPM, propeller thrust, motor current, or others.

This training data, comprises of features extracted from battery readings and all operational variables to be diagnosed as targets as shown in Eq. 1. Here, F contains feature vectors constructed from battery measurements and G represents a specific operational variable. The features data, F, and each target variable G_n, is used to train a machine learning model M_n resulting in $\{M_1, M_2, \ldots\}$. Unlike other approaches, the features data, in our approach, is the same for all operational variables. The trained machine learning models are to closely estimate different selected operational variables. A well trained model is not expected to predict the exact target variable values, but close enough so that the slope of the accumulated errors will follow a linear trend (He et al., 2019). The trained model also has to be computationally simple enough to enable real-time operations on available memory and processing power. This may be a tight constraint for on-board applications on UAVs or space systems. Different machine learning models may be trained and the one that suits the application may be selected.

$$F = \{f_1, \dots f_7\} \quad \text{and} \quad G = \{g_1, g_2, ...\} \tag{1}$$

3. DETECTING ANOMALIES USING RESIDUALS

For a given target variable g_n, model M_n gives a single prediction $\hat{G}_n^1$ from feature vector f_n^1. During normal operation, the predicted $\hat{G}_n^i$ should match its observed reading collected from the system, i.e., G_n^i. To check for anomalies, empirical readings $G_n = \{{g_n}_i^j\}$ are compared to the model estimated values $\hat{G}_n = \{{\hat{g}_n}_i^j\}$. An anomaly is detected based on the magnitude of deviation between the two, and not based on any pre-generated database of fault signatures or behaviors. Hence, in this unsupervised approach, the models are trained only on non-anomalous operational data. As a consequence, this approach is not restricted by the types of anomalies or just to known anomalous patterns. Instead, BDAV detects any new behavior that deviates from predicted norms.

3.1. Cumulative Residuals

To quantify how target predictions, $\hat{G}$, deviate from empirical readings, G, we use a summation of residuals over a specified time domain to compute a Cumulative Error Rate (CER) defined as:

$$e_i = ||G^i - \hat{G}^i|| = \sum_{j=1}^{w} \sqrt{\left(\hat{g}_j^i - g_j^i\right)^2} / g_j^i \times 100\% \tag{2}$$

The thus-calculated e_i considers each residual within the specified time window, effectively dampening the immediate effects of large variances in individual readings. If the time window considered is reduced down to a single reading, Eq. 2 may generate a wildly fluctuating plot with large changes from one reading to the next. On the other hand, increasing the window size to the range of a full test/operation will give only one error rate value. BDAV is based on detecting changes within this error rate. Hence the number of readings considered for each error calculation must lie somewhere between these two extremes where an anomaly to be diagnosed can significantly alter the CER value while the noise in measurements should be averaged out. We refer to the number of readings evaluated per error calculation as the window size, w. The w used in our error calculation marks the first of five configuration parameters that must be considered in our anomaly detection methodology. It is to be noted here that this window is different from the moving window utilized in the data pre-processing step to extract features from battery data. Anomalies can be detected based on the CER value whenever it breaches a pre-determined threshold for each operational variable. However, this does not consider the fact that some variables are more dynamic than others, and it also lacks methods to tune the models to achieve certain performance metrics (detection rate and false positive rates) which may be set by the end user according to the application. To add this sophistication, a few model parameters are introduced and are discussed below.

3.2. Error Weights

Equation 2 assigns equal weights to each reading within the time window considered. This equally dampens the impact a single large discrepancy can have, as it only accounts for $1/w$ of the total CER value. However, for certain short-duration anomalies, giving each reading within the time window equal weight can overly dampen short term error bursts. To increase the ability to capture such short error bursts, we add a weight coefficient to the most recent residual as

$$e_w = (b \cdot w) \sqrt{\left(\hat{g}_w - g_w\right)^2} / g_w \tag{3}$$

where b is between 0 and 1. Now, the most recent reading's effect increases to $b \times w$ the previous value. This increases the impact short but large error bursts can have on the total CER value. In addition to increasing the probability of capturing short anomalies, this reduces detection latency. The weight, b, given to the most recent reading in the CER calculation marks the second anomaly detection configuration parameter.

3.3. Dynamic Threshold

If the trained ML models are perfect, the residuals between the predicted and observed values will be zero. In reality, ML predictions are not perfect, and discrepancies exist between the predicted and the measured values. However, during non-anomalous operations, the mean of the residual values when predicting correlated operational variables are found to be consistent (He et al., 2019). This is apparent from plots showing cumulative error readings which exhibit linear trends. Our anomaly detection strategy is built around this observation. For a system's' operational variable, once the expected slope of the cumulative error plot is established, a significant change in its value indicates a potential anomaly.

Intuitively, detecting these changes could be done by simply monitoring for relative changes in the slope/derivative of the cumulative error readings in a moving window. However, this may not account for gradual, but consistent error changes over time and may also incorrectly classify short but sharp changes in error rate as anomalies. Instead, we classify anomalies using the calculated CER value by considering the number of such values that differ significantly from their previous values over a period of time. We increment a warning counter under the condition:

$$|\left(e_i - e_{i-1}\right)| > AvgError\left(e_i\right) + c\left(stddev\left(e_i\right)\right) \tag{4}$$

where e_i and e_{i-1} represent consecutive error readings. $AvgError$ is the average error for the current time window being examined for reading i. $stddev(e_i)$ is the standard deviation of readings over the time window. The coefficient c determines how much the error $stddev$ is factored into the threshold to consider the error change as anomalous. A smaller c effectively makes the model more sensitive to error rate changes and more likely to consider a slight increase in error rate as abnormal. Conversely, a larger c makes the model less sensitive to changes. This variable, c, marks the third configuration parameter in our anomaly detection configuration. The dynamic threshold, to increment the warning counter, is a result of considering the varying average error and the standard deviation values in Eq. 4. This dynamic threshold makes our anomaly detection sensitive to short anomalies while also minimizing false positives as a result of gradual wear and tear of components.

3.4. Warning Counter

As described above, each breach of the dynamic threshold increments a warning counter, t, rather than immediately flagging an anomaly. This prevents the algorithm from detecting a large number of false positives. A time frame from which error rates are considered to increment the warning counter is not defined.

Once a CER change is found to be above the computed warning threshold, the counter t is increased by one. When this occurs sufficient number of times for t to reach a certain limit, the system is considered to have shown to have a large enough discrepancy between predicted and measured values for a sufficient period of time, and an anomaly is flagged.

As BDAV is built to detect changes in behavior both instantaneous and also over time, warning counter increments do not need be consecutive. However, if these warning counts are allowed to continuously build up throughout a flight, eventually an false positive anomaly would be detected by the algorithm. To solve this issue, a decay value, as a percentage of the window size w, that decrements the warning counter is used. Once a CER value is found to be above the warning threshold the counter t is incremented, and when it is below the threshold, the decay, d, is incremented. Once the decay value reaches a certain limit (as a percentage of the window size w), t is decremented by one. By using a counter variable along with an associated decay, we ensure that anomalies are classified only for significant and consistent changes in cumulative error rates. The threshold of this warning counter, t, set as a percentage of window size, w, marks the fourth configuration parameter used in our anomaly detection.

3.5. Slope Detection

An important challenge when detecting anomalies lies in reducing the false detection/positive rate, i.e., when the detection model incorrectly classifies normal behavior as an anomaly. The tuning parameters mentioned above for the CER detection method determine both the number of true anomalies detected and that of false positives generated. A high anomaly detection accuracy can easily be achieved by configuring the CER model parameters to be overly sensitive to changes in error. However, this has a byproduct of a high false positive rate. Fine tuning of the configuration parameters is expected to increase detection rate and also reduce the false positive rate. In order to further reduce the number of false positives, we add an anomaly verification method based on changes in the slope of the error rate. This slope detection method utilizes the calculated CER as well as the same count threshold and decay parameters. However, instead of using the threshold formula Eq. (4) to increment threshold counts, the slope method compares the slope of first half to that of the second half of a considered time window w_i. The threshold count for this method, t_s, is incremented under the condition:

$$slope\left(e_{i-w/2}^{i}\right) > s \times slope\left(e_{i-w}^{i-w/2}\right). \qquad (5)$$

Table 1. Five key anomaly detection model configuration parameters

Parameters	Symbol
Window size	w
Error weight	b
Standard deviation coefficient	c
Slope coefficient	s
Warning counter threshold	t

The slope constant, s, used Eq. 5 to determine how severe the change in slope over a period of time must be in order to increment the warning counter for the slope based method. This slope detection method serves to validate anomalies detected using the dynamic threshold method. An anomaly is only classified when both methods detect an anomaly within a certain distance of each other, $(1.5 \times w)$. Slope coefficient s used in this method marks the fifth and final configuration parameter used in BDAV. The five key parameters are listed in Table 1 and the BDAV framework is illustrated in Fig. 1.

4. DEMONSTRATION ON AN ELECTRIC-PROPULSION TESTBED

4.1. Electric-Propulsion Testbed

A benchtop commercial-off-the-shelf testbed was assembled to demonstrate anomaly detection using the BDAV framework. This RC-Benchmark testbed is popular to test UAV electric propulsion systems (batteries, ESCs, motors, and propellers). The motor testbed was composed of a RCbenchmark Series 1580 Test Stand, BGNing 2212 2200kv brush-less motor, and a Zeee 11.1V 50C 3S lipo battery. The testbed is shown in Fig. 2, and has built-in load cells and other sensors

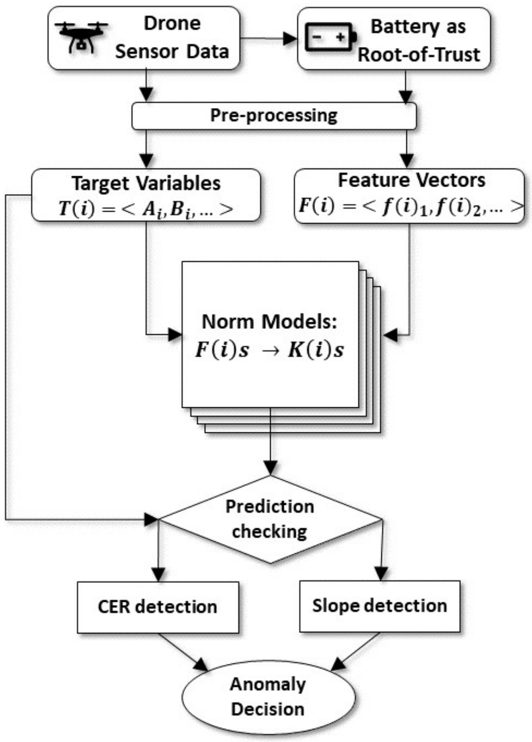

Figure 1. BDAV anomaly detection framework

to measure thrust, torque, motor RPM, vibration, voltage, current, and optional pins for temperature measurements.

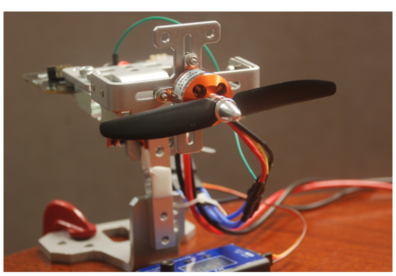

Figure 2. Motor-propeller testbed used for anomaly detection on motor RPM, thrust, & vibration measurements

This testbed was set up, and sample traces, at 50 Hz data logging frequency, were collected that represent normal system behavior. The test lasted 30-minutes while the throttle was manually and randomly varied. Sample data from this test, for battery current, motor RPM, and thrust measurements, are scaled to fit in the figure shown in Fig. 3. The physical relationships and correlations between the three plotted operational variables are apparent in the figure, which BDAV is designed to exploit to perform anomaly detection. We trained three norm machine learning models using gradient boosting algorithm, one for each of thrust, RPM, and vibration to predict their values using features extracted from battery voltage and current data. Then, we injected different types of anomalies described below to test and optimize the five tuning parameters described above.

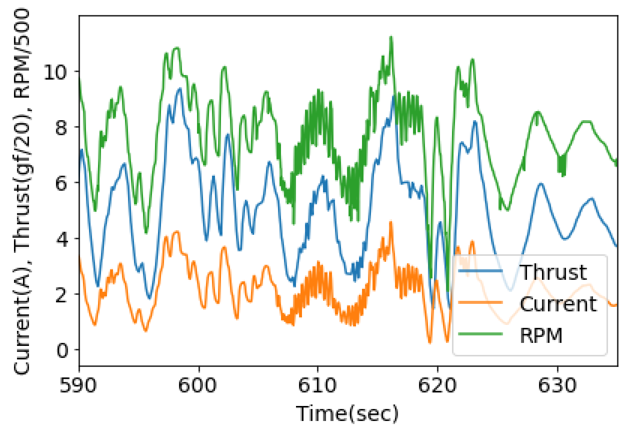

Figure 3. Normal operation data from testbed showing scaled current, thrust, & RPM measurements

4.2. Anomaly Injection

We modelled three types of anomalies injected randomly into the three considered operational variables. These anomalies relate to situations where 1. the measured sensor data deviates from the real observed behavior of the vehicle or its components (Yong, Yuanpeng, Yaqing, Yu, & Datong, 2017) or 2. when actuators do not respond according to their control inputs (Titouna, Naït-Abdesselam, & Moungla, 2020) due to actuator failures or control input corruption during a cyber attack. To simulate these faults, we substituted random lengths of the collected data at random locations with modified data. The three types of anomalies used to evaluate BDAV include (1) value shift, (2) random fluctuations, and (3) dropped signal as shown in Fig. 4

Anomaly Type I – Value Shift: The first anomaly type injected is a shift in the values $g(t)$ measured by a sensor. The values are randomly shifted using a constant percentage modifier, a, and modelled as:

$$G'(t) = G(t)(1 + a), \tag{6}$$

The value of a was randomly selected from a range (c, d) for each different anomalies injected into the data. This range modified the data by 20% to 80% of the original value. In addition to these randomized percentage modifiers, we also fixed a to specific values in order to determine the performance of BDAV models for different anomaly magnitudes. A shift in values of a measured operational variable like thrust or motor RPM could be due to a degraded sensor, motor or propeller. Increase in cumulative error, that can be detected using

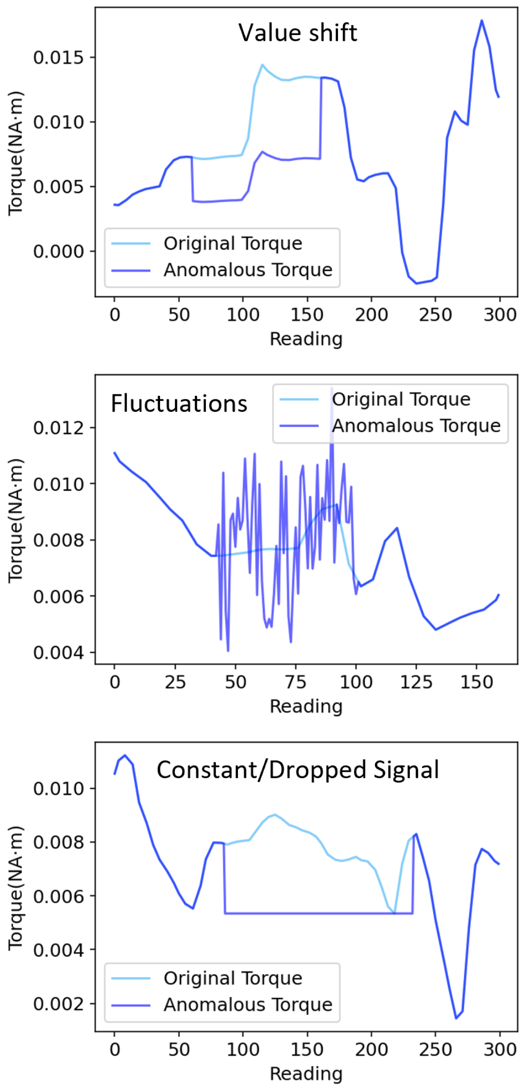

Figure 4. Three types of anomalies used to evaluate BDAV

BDAV framework, due to a value shift anomaly is shown in Fig. 5.

Anomaly Type II – Random Fluctuations: The second anomaly type injected was randomized fluctuations about the original data. For the duration of these anomalies, each individual true value $g(t)$ is multiplied by a random factor from a selected range, and is modelled as:

$$G^{\text{‘}}(t) = G(t) \times rand(1 - i, 1 + i), \quad (7)$$

where for each t during the anomaly, i is randomly selected from (c, d). Anomalies of this type emulate erratic sensor behavior due to noise or a failed sensor.

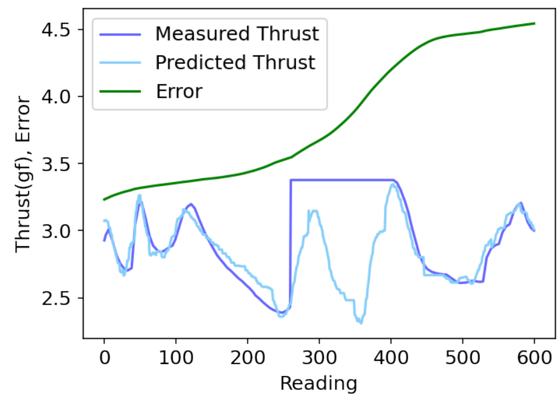

Figure 5. Increase in CER (green line) slope due to a value shift anomaly

Anomaly Type III – Dropped Signal: In the third anomaly type, measurements do not change for random durations. These are modelled by having a constant operational variable value (last true value measured) during the course of the anomaly. This may be indicative of a sensor intermittently failing to function as designed. During the anomaly, instead of the last measured true value, the constant value could also be set to 0 or any other fixed value as seen during sensor failures.

For each run evaluating BDAV, the lengths, injection points, strengths, and types of anomalies were randomized for each of the three target variables independent of each other. Anomaly lengths ranged from 30-180 readings (roughly 1-4 seconds in duration), with a randomized distance between each injected anomaly ranging from 50-400 readings. BDAV is expected to detect a variety of anomalies even with unknown behaviors. By injecting anomalies of varying types, strengths, durations, and locations, we are able to evaluate overall performance averages for detection rates, false positive rates, and detection latency.

4.3. Optimization of Key Parameters

Once the three types of anomalies are randomly injected into the dataset, one set of the five key parameters characterizing the anomaly detection algorithm, i.e., $\{w, b, c, s, t\}$ is selected and tested. Anomaly detection performance metrics such as detection rate, false positive rate, and detection latency are measured. This is repeated for other sets of the five key parameters. Instead of randomly/manually setting their values, a grid search was performed for each of the parameters to find the optimal set for the electric propulsion testbed. Changing each of the key parameters can have large impacts on detection rates, detection latency, and runtimes for anomaly detection. In addition to testing different sets of key

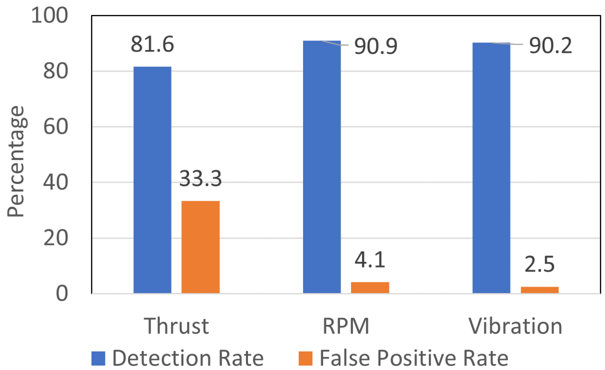

Figure 6. Anomaly detection performance on electric propulsion testbed's thrust, RPM, and vibration

parameters on the one dataset, multiple test datasets can also be generated with a different set of injected anomalies.

We tested window sizes ranging from 10 - 400 readings, error weights from 0% - 200%, stddev coefficients from 0.2 - 3, slope coefficients from 0.2 - 3, and threshold counts from 5% - 70% of the window size. Through this optimization study, best key parameter set was selected that resulted in detection and false positive rate for the three operational variables as shown in Fig. 6. Maximum anomaly detection rate was found to be 90.9% for motor RPM and the least false positive rate of 2.5% was detected in vibration measurements. Average anomaly detection latency was found to be only 1.9 seconds and these performance metrics are expected to further increase through additional testing and fine tuning of the key parameters.

Tuning of these parameters through repeated testing, while changing the parameters, is needed to adapt BDAV to each new platform. Trade offs in the performance metrics, as a result of the values of the five key parameters, are to be evaluated which should inform the selection of those values. This is expected to be a platform and application specific decision as a lower false positive rate may be more desirable in some situations than capturing all potential anomalies. The ideal configuration of BDAV is therefore dependent on user needs.

5. CONCLUSION

In this paper, we have proposed a new unsupervised machine learning approach, BDAV, to anomaly detection in battery powered vehicles, and demonstrated it on a eVTOL propulsion testbed. BDAV utilizes a system's battery as ground truth to detect even previously unseen anomalies during operations. It uses non-anomalous operational data to learn correlations from system's battery current and voltage readings independently to each system operational variable. Traditional machine learning models are utilized to generate one-one maps to predict values such as motor RPM, thrust, and others using only battery information. Summation of residual errors between predicted and observed values generally follows a linear trend, and any significant deviation, as characterized by five key parameters, is flagged as an anomaly.

This approach is demonstrated on an eVTOL propulsion testbed's thrust, motor RPM, and vibration measurements through a preliminary optimization of key parameters. Results showed anomaly detection rates as high as 90.9%, false positive rate as low as 2.5%, and an average detection latency of 1.9 seconds. This method is widely applicable to many platforms for onboard as well as off-board fault detection and identification. Root cause identification is a result of the one to one prediction and anomaly detection models that allows for simultaneous identification of possible root cause candidates. This work will work as the basis to demonstrate BDAV on a functional electric aerial vehicle and its many operational variables.

ACKNOWLEDGMENT

This work is funded by a NASA SBIR grant with contract No. 80NSSC21C0356. We would like to thank our technical monitor Mr. Ryan Mackey, for his guidance, and also several NASA personnel, for their valuable inputs during this project.

NOMENCLATURE

$BDAV$	battery-based diagnostics for aerial vehicles
$eVTOL$	electrical vertical take-off and landing
UAV	unmanned aerial vehicle
UAM	urban air mobility
$BLDC$	Brush-Less Direct Current
a	amplitude
t	timestamp
F	feature vectors from battery measurements
G	time series system data
CER, e	cumulative error rate
w	window size
b	weight parameter
c	error standard deviation parameter
t	warning counter parameter
s	slope parameter
RPM	rotations per minute
ESC	electronic speed controllers

REFERENCES

Bai, F., ElBatt, T. A., Holland, G., Krishnan, H., & Sadekar, V. (2006). Towards characterizing and classifying communication-based automotive applications from a wireless networking perspective. In *Autonet'06*.

Baker, C. R., Ferguson, D., & Dolan, J. M. (2008, July). Robust mission execution for autonomous urban driving. In *Ias'08* (p. 155-163).

Bauranov, A., & Rakas, J. (2019). Urban air mobility and manned evtols: safety implications. In *2019 ieee/aiaa 38th digital avionics systems conference (dasc)* (pp. 1–8).

Bemis, S. P., Micklethwaite, S., Turner, D., James, M. R., Akciz, S., Thiele, S. T., & Bangash, H. A. (2014). Ground-based and uav-based photogrammetry: A multi-scale, high-resolution mapping tool for structural geology and paleoseismology. *Journal of Structural Geology*, *69*, 163–178.

Chakraborty, A., Chai, E., Sundaresan, K., Khojastepour, A., & Rangarajan, S. (2018). Skyran: A self-organizing lte ran in the sky. In *Conext'18.*

Cho, K.-T., & Shin, K. G. (2016). Error handling of in-vehicle networks makes them vulnerable. In *Ccs'16.*

Choi, W., Jo, H. J., Woo, S., Chun, J. Y., Park, J., & Lee, D. H. (2016). Identifying ECUs Using Inimitable Characteristics of Signals in Controller Area Networks. *ArXiv e-prints*.

Diem, W. (2001). Smart card opens the door. *AutoTechnology*, *1*(1), 32–33.

Ebeid, E., Skriver, M., & Jin, J. (2017). A survey on open-source flight control platforms of unmanned aerial vehicle. In *2017 euromicro conference on digital system design (dsd)* (pp. 396–402).

ElBatt, T., Goel, S. K., Holland, G., Krishnan, H., & Parikh, J. (2006). Cooperative collision warning using dedicated short range wireless communications. In *Vanet'06.*

Feser, M., McConnell, D., Brandmeier, T., & Lauerer, C. (2006). Advanced crash discrimination using crash impact sound sensing (CISS). In *Sae technical paper.*

He, L., Kong, L., Liu, Z., Shu, Y., & Liu, C. (2019). Diagnosing vehicles with automotive batteries. In *Mobicom'19.*

Jones, W. D. (2002). Building safer cars. *IEEE Spectrum*, *39*(1), 82-85.

Lanigan, P. E., Kavulya, S., Narasimhan, P., Fuhrman, T. E., & Salman, M. A. (2011). *Diagnosis in automotive systems: A survey.*

Miller, C., & Valasek, C. (2015). A survey of remote automotive attack surfaces. In *Black hat usa.*

Mitchell, T. M., & Mitchell, T. M. (1997). *Machine learning* (Vol. 1) (No. 9). McGraw-hill New York.

Moradi, M., Sundaresan, K., Chai, E., Rangarajan, S., & Mao, Z. M. (2018). Skycore: Moving core to the edge for untethered and reliable uav-based lte networks. In *Mobicom'18.*

Murvay, P. S., & Groza, B. (2014). Source identification using signal characteristics in controller area networks. *IEEE Signal Processing Letters*, *21*(4), 395-399.

Pesé, M. D., Ganesan, A., & Shin, K. G. (2017). Carlab: Framework for vehicular data collection and processing. In *Carsys'17.*

Quinones-Grueiro, M., Biswas, G., Ahmed, I., Darrah, T., & Kulkarni, C. (2021). Online decision making and path planning framework for safe operation of unmanned aerial vehicles in urban scenarios. *International Journal of Prognostics and Health Management*, *12*(3).

Silva, C., Johnson, W. R., Solis, E., Patterson, M. D., & Antcliff, K. R. (2018). Vtol urban air mobility concept vehicles for technology development. In *2018 aviation technology, integration, and operations conference* (p. 3847).

Swartz, K. I. (2017). Charging forward: New evtol concepts advance. *Vertiflite*, *4*, 24–29.

Thiels, C. A., Aho, J. M., Zietlow, S. P., & Jenkins, D. H. (2015). Use of unmanned aerial vehicles for medical product transport. *Air medical journal*, *34*(2), 104–108.

Titouna, C., Naït-Abdesselam, F., & Moungla, H. (2020). An online anomaly detection approach for unmanned aerial vehicles. In *2020 international wireless communications and mobile computing (iwcmc)* (pp. 469–474).

Tran, M.-K., & Fowler, M. (2020). A review of lithium-ion battery fault diagnostic algorithms: Current progress and future challenges. *Algorithms*, *13*(3), 62.

Velusamy, P., Rajendran, S., Mahendran, R. K., Naseer, S., Shafiq, M., & Choi, J.-G. (2021). Unmanned aerial vehicles (uav) in precision agriculture: applications and challenges. *Energies*, *15*(1), 217.

Waraksa, T. J., Fraley, K. D., Kiefer, R. E., Douglas, D. G., & Gilbert, L. H. (1988). *Passive keyless entry system* (Nos. US, 5,319,364A).

Wasicek, A., Pese, M. D., Weimerskirch, A., Burakova, Y., & Singh, K. (2017). Context-aware intrusion detection in automotive control systems. In *Escar'17.*

Yong, D., Yuanpeng, Z., Yaqing, X., Yu, P., & Datong, L. (2017). Unmanned aerial vehicle sensor data anomaly detection using kernel principle component analysis. In *2017 13th ieee international conference on electronic measurement & instruments (icemi)* (pp. 241–246).