

Anomaly detection in micro air vehicle link telemetry logs of unmanned aerial vehicles using data analytics methods

Paweł Dymora^{1*}, Mirosław Mazurek¹, Bartosz Nycz²

¹ Faculty of Electrical and Computer Engineering, Rzeszów University of Technology, al. Powstańców Warszawy 12, 35-959 Rzeszów, Poland

² Państwowa Akademia Nauk Stosowanych w Przemyślu, Poland

* Corresponding author's e-mail: Pawel.Dymora@prz.edu.pl

ABSTRACT

Unmanned aerial vehicles rely on telemetry streams for mission monitoring, diagnostics, post-flight analysis, and cyber-physical security assessment. This paper presented a data analytics workflow for anomaly detection in micro air vehicle link (MAVLink) telemetry logs. The method combines protocol-aware preprocessing, time-series feature engineering, interpretable rule-based indicators, replay-like communication-pattern screening, and unsupervised anomaly detection. The analysed variables include position, velocity, attitude, altitude, battery parameters, system-status messages, message timing, and selected communication patterns. The workflow was evaluated in complementary stages. First, it was applied to publicly available UAV logs from DroneKit-LA and representative annotated cases from UAV-SEAD, where detected anomaly windows were interpreted against diagnostic outputs, status messages, and available log-level annotations. Second, controlled MAVLink-like validation fixtures were used to verify detector behaviour under known anomaly conditions and to quantify sample-level precision, recall, F1-score, and ranking ability. The public-log analysis showed that the proposed indicators can highlight the windows consistent with altitude-estimate divergence, attitude-control problems, estimator warnings, compass/variance flags, and communication irregularities. In the five-flight validation fixture, rule-based indicators achieved high precision but lower recall, while Isolation Forest and One-Class SVM provided stronger anomaly-score ranking at the cost of more false positive detections. The findings show that lightweight unsupervised models are useful for ranking suspicious windows, whereas domain rules provide clearer operational interpretation. The proposed workflow should be understood as an interpretable screening and post-flight diagnostic pipeline, rather than a complete operational intrusion-detection system.

Keywords: UAV, MAVLink, telemetry logs, anomaly detection, data analytics, replay attack, flight log analysis, cybersecurity.

INTRODUCTION

Unmanned aerial vehicles (UAVs) are increasingly used in inspection, infrastructure monitoring, precision agriculture, logistics, rescue operations, and research applications. Their safe operation depends not only on airframe reliability and flight-control algorithms, but also on continuous exchange of telemetry and command data between the aerial platform and the ground control station (GCS). MAVLink is one of the most widely used communication protocols in open

UAV ecosystems, especially in the systems based on ArduPilot and PX4. Due to its low overhead and suitability for resource-constrained communication links, MAVLink is commonly used to transmit flight state, position, attitude, battery status, mission information and command messages.

Telemetry logs provide a valuable source of evidence for analysing UAV behaviour. Similar data-driven analysis pipelines have become increasingly important in other engineering measurement systems, where derived features provide significantly more diagnostic information

than raw sensor signals [1, 2]. They make it possible to reconstruct the flight trajectory, identify abnormal sensor readings, correlate system messages with physical events, and detect suspicious communication patterns. However, telemetry recorded at GCS represents only the data that successfully passed through the radio link. Therefore, MAVLink.tlog files may include packet losses, irregular sampling, retransmissions, and incomplete information about high-frequency onboard processes. These limitations make anomaly detection in the MAVLink telemetry logs a challenging data analytics problem and increase the risk of false positive detections when communication artefacts are interpreted as physical anomalies.

The existing research on UAV anomaly detection has proposed a wide range of methods, including online subspace tracking and mitigation [3] as well as surveys of flight-data anomaly detection [4]. Spatio-temporal recovery models represent another important methodological group [5]. Nevertheless, many studies focus mainly on sensor faults or general flight-data anomalies. Less attention is given to practical workflows that jointly analyse operational anomalies and communication-level security anomalies in the MAVLink telemetry logs. This is important because cyber-physical incidents may manifest themselves not only as abnormal packet patterns, but also as unexpected changes in trajectory, speed, attitude, battery state or failsafe events.

The aim of this paper was to develop a reproducible data analytics workflow for detecting anomalies in MAVLink telemetry logs of UAV systems. The proposed approach combines domain knowledge, time-series feature engineering, rule-based detection, replay-like communication-pattern screening, and unsupervised machine learning. The main contributions of the paper are: a protocol-aware preprocessing and feature-engineering workflow for MAVLink telemetry logs; a hybrid detection strategy combining interpretable rule-based checks with unsupervised anomaly scoring; a real-log validation procedure in which the detected windows are compared with public diagnostic outputs and annotated cases; a controlled validation fixture for repeatable benchmarking of detector behaviour under known anomaly conditions; and an event-level interpretation framework linking detected anomalies with UAV mission safety and cybersecurity.

RELATED WORKS

Flight-data anomaly detection is a central problem in the safety analysis of UAV systems. Telemetry streams include strongly interdependent variables: position, velocity, altitude, attitude, battery parameters, mode changes and system messages. For this reason, the literature increasingly treats UAV logs as multivariate time-series data, rather than isolated sensor measurements [4, 5]. Such logs are especially valuable for post-flight diagnostics, incident analysis and security auditing, but their interpretation requires domain knowledge and careful treatment of missing or asynchronous samples.

The survey by Yang et al. [4] provided a broad overview of technologies, applications, and future research directions in UAV flight-data anomaly detection. The authors showed that methods differ in terms of required training data, computational cost, interpretability, and ability to recover corrupted signals. This observation is important for MAVLink analysis because the telemetry logs recorded by GCS are often incomplete and may not fully reflect high-frequency onboard behaviour.

Early data-driven approaches focus on detecting deviations in multivariate flight data using statistical distance, regression models or subspace representations. He et al. [3] proposed ADMOST, a method for UAV flight-data anomaly detection and mitigation via online subspace tracking. The relevance of ADMOST lies in its combined detection and mitigation logic: the method not only identifies abnormal observations, but also supports recovery of abnormal flight data. This is important for safety-critical systems, where diagnostics should support decision-making rather than only generate alerts.

Another important research direction is based on spatio-temporal correlation. Zhong et al. [5] proposed an LSTM-based approach for anomaly detection and recovery prediction using spatio-temporal relationships in UAV flight data. Yang et al. [6] developed a data-driven unsupervised method based on spatio-temporal correlation and LSTM autoencoders. Related multivariate and multiple-regression frameworks for anomaly detection and recovery were later proposed by Yang et al. [7, 8]. These studies show that the UAV telemetry parameters cannot be analysed independently: position, velocity, altitude, orientation and control state form a coupled

cyber-physical process. Unsupervised learning is widely used when labelled UAV anomalies are rare or expensive to obtain. Sezgin et al. [9] proposed an automated machine learning approach for UAV telemetry anomaly detection and evaluated unsupervised algorithms such as Local Outlier Factor, Isolation Forest, One-Class SVM and Elliptic Envelope. This work is particularly relevant to the present article, because it supports the use of lightweight, interpretable machine learning methods for telemetry logs where labelled anomalies may be unavailable.

Deep learning methods have also been applied to UAV anomaly detection. Bell et al. [10] used a stacked recurrent autoencoder for UAV sensor-data anomaly detection and dynamic thresholding. Zhou et al. [11] combined wavelet decomposition with a stacked denoising autoencoder to improve robustness under noisy conditions. Chen et al. [12] proposed a convolutional autoencoder combined with support vector data description and soft-margin loss. A recent systematic review also confirmed the rapid growth of deep learning-assisted UAV flight-data anomaly detection [18]. These methods provide high modelling capacity and can capture nonlinear temporal patterns, but they require sufficient training data, careful calibration and more computational resources than rule-based methods.

Recent work also explored graph-based and causality-aware representations. Feng et al. [13] proposed causality-enhanced graph neural networks for UAV anomaly detection, while Feng et al. [14] introduced the STE-RS framework for robust anomaly detection in UAV flight data. These approaches are promising because telemetry variables are naturally structured as a network of causal and temporal dependencies. However, for practical MAVLink post-flight analysis, such models should be complemented with interpretable indicators that can be understood by operators and forensic analysts.

A separate branch of the literature addressed drone flight logs from a forensic and severity-classification perspective. Silalahi et al. [15] used transformer-based sentiment analysis for anomaly detection in drone forensic logs. Later, Silalahi et al. [16] proposed a severity-oriented multiclass approach for drone flight-log anomaly detection. Silalahi et al. [17] introduced an interpretable ordinal-aware and contrastive-enhanced framework for detecting anomaly severity in UAV logs. These studies are relevant because practical

post-flight analysis should not only decide whether an event is anomalous, but also estimate its operational or security severity.

The present study followed this logic by separating operational anomalies from communication-level security anomalies. Operational anomalies include physically inconsistent GPS movement, excessive vertical velocity, sudden voltage drops, and failsafe messages. Communication-level anomalies include replay-like packet patterns, duplicated commands, irregular message intervals and sequence inconsistencies. This distinction helps reduce ambiguity in interpreting anomaly scores.

Replay attacks are especially relevant to UAV communication because an attacker may reuse the previously captured packets. In a simple case, replay-like behaviour may appear as repeated command messages, duplicated payloads, abnormal message timing, or inconsistent command-state relationships. However, a capable attacker may modify sequence numbers, alter timing, or replay only selected command fields. Therefore, sequence repetition alone is not sufficient for real-world replay detection. The present workflow treats replay detection as a set of replay-like indicators, rather than as cryptographic proof of attack.

The cybersecurity dimension of UAV telemetry analysis therefore requires more than classical sensor-fault detection. It should include packet-level and message-level indicators, including inter-arrival times, duplicate-message patterns, sequence discontinuities, abnormal command frequency, command-state consistency, and acknowledgement behaviour. A hybrid approach combining domain rules, statistical indicators, and unsupervised learning is appropriate because it can capture both known patterns and unexpected deviations while retaining operational interpretability.

MATERIALS AND MODELS

MAVLink logs and preprocessing

The proposed study used the MAVLink telemetry logs recorded by a ground control station. Such logs contain the communication stream exchanged between the UAV and the GCS, including telemetry messages, selected state variables, status information and operator commands. The relevant MAVLink message

groups include: HEARTBEAT, SYS_STATUS, GPS_RAW_INT, GLOBAL_POSITION_INT, LOCAL_POSITION_NED, ATTITUDE, VFR_HUD, STATUSTEXT and selected command messages. Depending on logging configuration, additional messages may describe radio status, mission state, servo outputs or battery state.

A key limitation of the .tlog data is that the log reflects the radio-link perspective. If packets are lost or delayed, the ground-station log may contain gaps or irregular timestamps. Therefore, preprocessing must normalise time, handle missing data and avoid interpreting every discontinuity as a flight anomaly. The proposed workflow treats communication irregularities both as possible artefacts and as potential security indicators requiring correlation with other signals.

The preprocessing stage converts raw MAVLink messages into synchronised time-series tables. The main operations include message selection, timestamp normalisation, removal of incomplete records, unit conversion, outlier filtering for impossible values, resampling to a common time grid and construction of analysis windows. For example, geographic coordinates are converted into metric displacement, velocities are checked for physical consistency and attitude angles are normalised to a common representation. Missing values are handled using short-gap interpolation only when physically justified. Longer gaps are preserved as separate communication-quality indicators.

Feature engineering

Feature engineering transforms raw telemetry fields into the variables that capture operational state and communication behaviour. Flight-dynamics features include horizontal speed, vertical speed, acceleration proxies, angular-rate changes, altitude-rate changes, distance between consecutive GPS samples, and battery voltage-drop rate, as shown in Table 1. Communication-level features include message interarrival time, message frequency, duplicate payload indicators, repeated command patterns, sequence discontinuities, and abnormal distributions of message types. The feature set is designed to support both rule-based and model-based anomaly detection.

Feature engineering is a central methodological step in the proposed workflow. Raw MAVLink fields are not used only as isolated variables. They are transformed into diagnostic indicators

that express physical consistency, temporal continuity, energy behaviour, attitude dynamics, and communication regularity. This approach is similar to diagnostic feature extraction used in other measurement-intensive domains, such as industrial condition monitoring, network intrusion detection, vibration analysis, optical sensing, and energy-system diagnostics. Similar data-driven feature engineering and machine-learning-assisted processing have also been successfully applied in optical measurement systems and fibre-optic sensing for parameter estimation, calibration, and signal interpretation [1, 2]. In these domains, raw signals are often less informative than derived indicators, such as rates of change, residuals, temporal patterns, distribution shifts, or consistency checks between related variables.

In the UAV telemetry context, feature engineering connects numerical anomaly scores with operational interpretation. A raw GPS coordinate is transformed into displacement and implied speed. Battery voltage is transformed into voltage-drop rate. Attitude angles are transformed into angular-rate indicators. MAVLink command streams are transformed into message-frequency, sequence-consistency, and duplicate-payload features. This transformation improves interpretability and allows the same feature table to be used by both rule-based and model-based detectors.

Rule-based indicators and threshold calibration

Rule-based indicators provide interpretable baselines and are useful for the events that can be described using physical constraints or protocol-level logic. A GPS-jump indicator is triggered when the displacement between two consecutive geographic samples implies a velocity exceeding the expected capability of the UAV platform. A voltage-sag indicator is triggered when a sudden voltage drop occurs together with high current demand or a related status message. A failsafe indicator is triggered by specific status messages or mode changes. A replay-like indicator is triggered by duplicated commands, repeated payloads or abnormal timing patterns.

The thresholds used by rule-based indicators should not be interpreted as universal constants applicable to all UAV platforms (Table 2). They are platform- and mission-dependent parameters. In the proposed workflow, threshold selection follows a three-step procedure. First,

Table 1. Proposed MAVLink features and anomaly interpretation

Feature group	Example features	Possible anomaly	Interpretation
Position / GPS	Distance between consecutive GPS samples; implied ground speed; coordinate discontinuity	GPS jump; spoofing-like movement; multipath error	Physically unrealistic displacement or trajectory discontinuity.
Velocity	vx, vy, vz; total speed; acceleration proxy	Sudden speed spike; inconsistent vertical motion	May indicate an aggressive manoeuvre, sensor error or logging artefact.
Attitude	roll, pitch, yaw; angular-rate differences	Abrupt attitude change; unstable control	May indicate loss of stability, wind disturbance or control anomaly.
Battery/power	voltage; current; voltage-drop rate	Voltage sag; battery failsafe	May indicate high load, weak battery or false failsafe trigger.
System status	STATUSTEXT; failsafe flags; mode changes	Failsafe event; EKF or GCS loss warning	Provides a semantic explanation for physical anomalies.
Communication	inter-arrival time; duplicate payload; sequence gaps; repeated commands	Replay-like event; packet loss; abnormal message burst	Supports security-oriented interpretation of telemetry logs.

physically impossible or highly unlikely values are identified using platform specifications, such as expected maximum horizontal speed, maximum climb rate, battery configuration, and nominal current range. Second, robust statistics are calculated from normal or quasi-normal flight segments. In particular, upper quantiles and the median absolute deviation are used to define conservative anomaly thresholds. Third, the selected thresholds are checked against semantic diagnostic evidence, including STATUSTEXT messages, failsafe indicators, EKF warnings, and mission context.

For example, the horizontal-speed threshold is intended to detect physically unrealistic displacement between consecutive GPS samples, rather than aggressive but feasible manoeuvres. Therefore, it can be defined as the maximum of a platform-based limit and a robust data-driven limit: $T_{\text{speed}} = \max(1.5 \times V_{\text{max_platform}}, Q99.5(v_{\text{h_normal}}) + 3 \times \text{MAD}(v_{\text{h_normal}}))$. Similarly, the voltage-drop threshold can be

calibrated as $T_{\text{voltage}} = Q99.5(s_{\text{V_normal}}) + 3 \times \text{MAD}(s_{\text{V_normal}})$, while also considering the battery type, nominal voltage, expected current draw, and empirical distribution of voltage changes during normal flight.

Diagnostic indicators and model-based detection

The analytical layer converts raw MAVLink telemetry fields into physically interpretable indicators. Let t_i denote the timestamp of sample i , while ϕ_i and λ_i represent latitude and longitude. For short flight segments, local displacement may be approximated from consecutive geographic samples and then converted into implied horizontal speed. Table 3 summarises the diagnostic indicators used in the workflow.

For multivariate anomaly detection, a feature vector x_i is created for each time window. In the simulation, x_i includes speed, vertical speed,

Table 2. Rule-based thresholds and calibration logic

Indicator	Example threshold used in fixture	Calibration principle	Interpretation
GPS jump	horizontal speed > 35 m/s	Platform speed limit + robust upper quantile from normal data	Physically implausible displacement.
Vertical anomaly	platform-specific vertical-speed limit	Climb/descent capability + normal-flight distribution	Altitude-estimation or barometric/GPS inconsistency.
Voltage sag	voltage-drop rate > 0.35 V/s	Battery type, current draw, robust voltage-change statistics	Sudden power disturbance.
Attitude disturbance	roll/pitch/yaw-rate limit	Platform dynamics + robust angular-rate statistics	Abrupt control or attitude anomaly.
Replay-like command	duplicated payload or sequence irregularity in a time window	Protocol-level consistency and command-frequency baseline	Candidate communication anomaly.

roll-rate, pitch-rate, yaw-rate, voltage-drop rate, current change, and repeated-sequence indicator.

The model-based stage compares selected unsupervised anomaly detection methods. Isolation Forest is selected because it performs well on high-dimensional tabular data and identifies the observations that are easy to isolate. Local Outlier Factor is selected because it captures local density deviations. One-Class SVM is selected as a boundary-based method for learning the region of normal behaviour. Depending on data volume, an autoencoder or recurrent autoencoder can be used as an additional baseline, following recurrent, convolutional, and denoising-autoencoder approaches reported in the literature [10, 12]. Model outputs are expressed as anomaly scores assigned to time windows and then correlated with rule-based indicators.

The evaluation strategy combines quantitative comparison and expert interpretation. If labelled anomalies are available, precision, recall, F1-score, ROC AUC, and false-alarm rate are reported. If labels are unavailable, the evaluation uses manual inspection of top-ranked anomaly windows, agreement with rule-based indicators, and consistency with status messages. Runtime and memory consumption should also be reported when the method is transferred to real-time or edge-based UAV diagnostics [19].

Replay-like communication anomaly scenarios

Replay-related anomalies in MAVLink telemetry should be interpreted with caution. A simple replay may appear as repeated command payloads, duplicated sequence numbers, or abnormal

inter-arrival times. However, a more capable attacker may modify sequence numbers, alter timing, or replay only selected command fields. Therefore, sequence repetition alone is not sufficient to detect all replay scenarios. The proposed workflow treats replay detection as screening for replay-like command anomalies, rather than as a cryptographic proof of attack.

To broaden the detection logic, a replay-like analysis includes complementary indicators: duplicate or near-duplicate command payloads, abnormal command frequency, repeated command parameters outside the expected mission context, inconsistent command-state transitions, missing or duplicated acknowledgements, abnormal inter-arrival times, and disagreement between command messages and subsequent vehicle state. The replay-like communication scenarios considered in this study, together with their expected observable indicators, are summarised in Table 4. This broader interpretation helps distinguish malicious replay from benign retransmission, parser duplication, operator repetition, or radio-link instability.

Evaluation protocol and statistical treatment

The experimental material used in this study has three different roles. Public real UAV logs are used as the primary empirical source for checking whether the proposed workflow can identify suspicious windows consistent with diagnostic events or annotated anomaly categories. Drone-Kit-LA analyser outputs and UAV-SEAD annotations are used as external references for event-level interpretation, rather than outputs of the proposed detector. The generated five-flight fixture

Table 3. Diagnostic indicators used for anomaly detection in MAVLink telemetry logs

Indicator	Formula/condition	Interpretation
Horizontal displacement	$d_i = \sqrt{\Delta x_i^2 + \Delta y_i^2}$	Distance between consecutive telemetry samples
Horizontal speed	$v_h = d_i / \Delta t_i$	An unrealistic value may indicate a GPS jump or a corrupted position
Vertical speed	$v_z = \Delta h_i / \Delta t_i$	Abrupt value may indicate altitude anomaly
Attitude rate	$\omega_{roll} = \Delta roll_i / \Delta t_i$; $\omega_{pitch} = \Delta pitch_i / \Delta t_i$	Large angular rates may indicate instability or a logging artifact
Voltage-drop rate	$s_V = -\Delta V_i / \Delta t_i$	High value with a current spike may indicate a voltage sag
Sequence anomaly	$r_i = 1(\text{seq}_i \leq \text{seq}_{\{i-1\}})$	Possible duplicate, replay-like pattern, or logging disorder
Duplicate payload	$q_i = 1(\text{hash}(\text{payload}_i) \text{ already observed})$	Repeated message content in a sliding window
Replay-like score	$R_i = w_1 r_i + w_2 q_i + w_3 \text{robust}_z(r_i)$	Aggregated communication-level anomaly indicator

Table 4. Replay-like MAVLink communication scenarios and expected indicators

Scenario	Description	Expected indicators	Limitation
Naive replay	Previously captured command is resent without modification.	Duplicate payload, repeated sequence, abnormal timing.	Easy to detect but not representative of advanced attacks.
Sequence-renumbered replay	Payload is replayed with a modified sequence number.	Duplicate command parameters, abnormal timing, command-state mismatch.	Sequence-based rule may fail.
Timing-shifted replay	The old command is replayed later in a different mission phase.	Mission-context inconsistency, repeated payload, abnormal command timing.	Requires mission-context modelling.
Partial replay	Only selected fields or command parameters are reused.	Parameter similarity, abnormal command pattern.	Harder to distinguish from a legitimate operator action.
Benign retransmission / duplication	Repeated message caused by radio link, logging, or parser behaviour.	Duplicates without physical effect or state inconsistency.	Should not be classified as an attack without additional evidence.

is used only as a controlled and reproducible benchmark, because it provides known labels and enables calculation of precision, recall, F1-score, ROC AUC, and sensitivity measures. The initial single-flight simulation serves as a preliminary proof-of-concept and should not be interpreted as real-flight validation.

To assess statistical uncertainty in controlled benchmarks, sample-level metrics are accompanied by approximate 95% confidence intervals derived from the underlying counts of true positives, false positives, false negatives, and true negatives. Because UAV telemetry is a time-series signal and consecutive samples are not statistically independent, these intervals should be interpreted as indicative rather than definitive. For future repeated generated runs, event-level block bootstrap and paired non-parametric tests should be used to compare detectors across random seeds, anomaly magnitudes, anomaly durations, packet-loss rates, and model contamination parameters. The recommended parameter ranges and experimental settings for these sensitivity analyses are summarised in Table 5.

RESULTS

Real-log validation on public UAV data

The first stage of the evaluation was based on public UAV logs. Unlike the generated validation fixture, these logs were not created to match the proposed rules and therefore provide a more realistic test of the workflow. The proposed pipeline was applied directly to the available telemetry variables, and the resulting anomaly windows were compared with diagnostic events, analyser outputs, and, where available, dataset-level annotations.

A search of public data sources identified two immediately reusable ArduPilot test logs and one large-scale real-world PX4 dataset suitable for strengthening the empirical part of the study. ArduPilot distinguishes between the DataFlash logs stored onboard and the telemetry logs recorded by the ground station over a telemetry link [20]. This distinction is important because .tlog files reflect the radio-link/GCS perspective, whereas .bin/.ulg files preserve richer onboard measurements. The first public source is the DroneKit-LA

Table 5. Sensitivity-analysis parameters recommended for repeated generated runs

Parameter	Range/variants	Purpose
GPS-jump magnitude	20–80 m	Sensitivity to position-anomaly strength.
Voltage-drop rate	0.15–0.60 V/s	Sensitivity to power-event intensity.
Attitude-spike amplitude	5–25 degrees	Sensitivity to control disturbance.
Replay-segment duration	5–40 s	Sensitivity to communication-anomaly length.
Background telemetry noise	low / medium / high	Robustness to noisy telemetry.
Packet-loss rate	0%, 2%, 5%, 10%	Robustness to GCS logging artefacts.
Model contamination	0.01–0.10	Sensitivity of unsupervised methods.

test-data repository, which contains sample ArduPilot DataFlash and MAVLink telemetry logs and explicitly states that the files can be freely used for testing DroneKit Log Analyzer [21]. This source provides a real MAVLink telemetry file (flight.tlog) and a real ArduPilot DataFlash file (log171.bin), together with expected analyzer outputs. The second source is UAV-SEAD, a real-world PX4 state-estimation anomaly dataset containing categorised annotated flight logs and raw logs [22]. The TLM:UAV dataset is useful as an additional simulation benchmark, but it should not be described as real flight evidence [23].

For the DroneKit-LA MAVLink telemetry case, the proposed feature extraction identified the candidate windows associated mainly with altitude-related inconsistency and attitude-control behaviour. These windows are consistent with the diagnostic reference output, which reports altitude-estimate divergence and attitude-control test failure. Importantly, the DroneKit-LA output was not used to generate anomaly labels. It was used only after detection as an external reference for interpreting the detected windows.

For the DroneKit-LA DataFlash case, the detected anomaly windows were concentrated around estimator-related and attitude-related inconsistencies. These results are consistent with the reference diagnostic information indicating

altitude-estimate divergence, EKF-status problems, and compass/variance warnings. At the same time, battery-related indicators remained low, which agrees with the reference output reporting that the battery test passed. This supports the usefulness of the hybrid approach, because it does not merely assign a generic anomaly label to the entire flight but helps distinguish estimator-related events from power-related events.

For representative UAV-SEAD cases, the pipeline can be applied to annotated logs covering normal operation and selected anomaly categories. The real-log evaluation confirms that high anomaly scores are more informative when interpreted at the event level and correlated with physical or estimator-related indicators. In normal logs, isolated high scores may be associated with telemetry irregularities or short transient changes, whereas in anomalous logs, suspicious windows should be linked with position, altitude, attitude, or estimator-related features. Representative validation results obtained using the proposed anomaly detection pipeline are summarised in Table 6.

Initial single-flight controlled benchmark

To illustrate the proposed workflow before processing a complete real UAV campaign, a controlled MAVLink-like simulation was prepared. The

Table 6. Real-log validation results using the proposed anomaly detection pipeline

Case	Dataset	Log type	Reference information	Main indicators detected by proposed pipeline	Event-level interpretation
DKL-TLOG-01	DroneKit-LA	MAVLink .tlog	Altitude-estimate divergence; attitude-control test failure.	Altitude inconsistency, attitude-rate changes, selected high anomaly-score windows.	Detected windows are consistent with altitude and attitude-control problems.
DKL-BIN-171	DroneKit-LA	ArduPilot .bin	Altitude-estimate divergence; EKF-status problems; compass/variance warnings; battery test passed.	Estimator-related inconsistency, attitude/altitude deviations, low battery-anomaly indicators.	Method separates estimator-related anomalies from power-related events.
USEAD-NORMAL	UAV-SEAD	PX4 .ulg / extracted topics	Normal flight category.	Low concentration of rule-based anomalies; scattered unsupervised outliers.	Mostly normal operation with isolated telemetry irregularities.
USEAD-GPOS	UAV-SEAD	PX4 .ulg / extracted topics	External/global-position anomaly category.	Position and velocity inconsistency; high position-related anomaly scores.	Consistent with a global-position anomaly.
USEAD-ALT/MECH	UAV-SEAD	PX4 .ulg / extracted topics	Altitude or mechanical/electrical anomaly category.	Altitude, attitude, battery, or estimator-related indicators depending on the case.	Consistent with cyber-physical or estimator-related disturbances.

simulation is not intended to replace experimental validation. It provides a transparent benchmark in which the location and type of injected anomalies are known. This enables preliminary verification of feature engineering, rule-based indicators and unsupervised anomaly detection models.

The simulated dataset represents a short UAV mission with local coordinates, altitude, attitude, battery voltage, current, MAVLink-like message type, command field and sequence number. The injected anomalies were placed at known time intervals in order to test whether the proposed indicators can identify both operational and communication-level deviations, as shown in Table 7.

The simulated benchmark confirms that transparent rule-based indicators are useful when the expected anomaly has a clear physical or communication-level definition. In this scenario, the rule-based layer achieved the highest F1-score, mainly because the injected anomalies were designed to match interpretable domain conditions, as shown in Table 8. Among unsupervised models, Local Outlier Factor performed better than Isolation Forest and One-Class SVM, but all model-based methods produced more false positives than the rule-based layer.

The obtained numerical values should be treated as simulated preliminary results, rather than

Table 7. Injected anomaly scenarios and expected detector responses

Injected anomaly	Time/interval	Modified variables	Expected detector response
GPS jump	$t = 420$ s	north/east displacement	Unrealistic horizontal speed and trajectory discontinuity
Voltage sag	$t = 700\text{--}719$ s	battery voltage and current	High voltage-drop rate and battery-current inconsistency
Attitude spike	$t = 1010\text{--}1021$ s	roll and pitch	Large angular-rate features
Replay-like command segment	$t = 1210\text{--}1229$ s	COMMAND_LONG, repeated sequence	Repeated sequence and command-pattern anomaly

Table 8. Preliminary detection results for the initial single-flight benchmark

Method	Precision	Recall	F1-score	ROC AUC	Detected points
Rule-based indicators	0.953	0.774	0.854	n/a	43
Isolation forest	0.528	0.528	0.528	0.764	53
Local outlier factor	0.604	0.604	0.604	0.758	53
One-class SVM	0.364	0.377	0.370	0.746	55

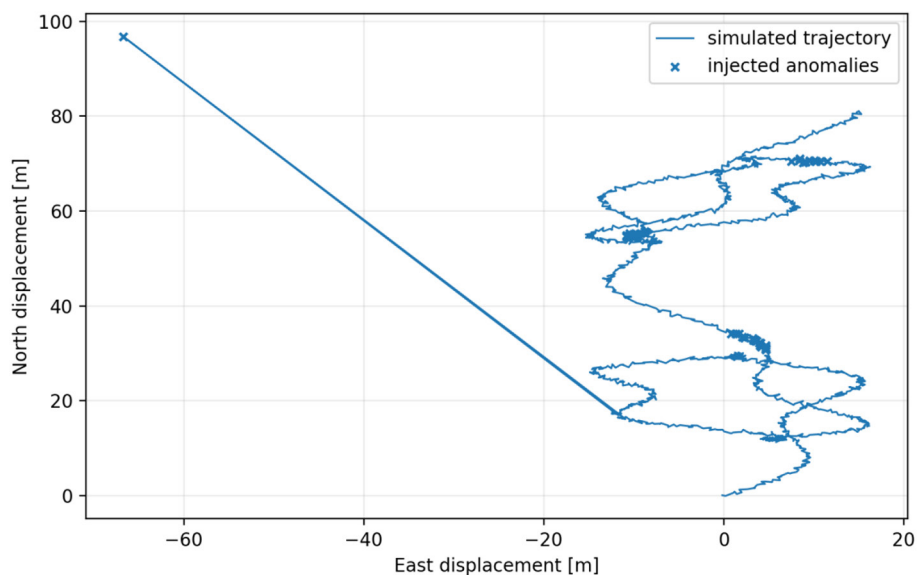


Figure 1. Simulated UAV trajectory with injected anomalies

final empirical evidence. The main purpose of this benchmark is to confirm that the feature-engineering and detection pipeline can identify the deliberately injected and clearly separated anomalies.

The spatial distribution of the simulated UAV mission is presented in Figure 1. The trajectory shows a generally smooth flight path, interrupted by deliberately injected anomaly points. The most visible disturbance corresponds to the GPS-jump scenario, where the position signal changes abruptly and creates a physically unrealistic displacement between consecutive telemetry samples. This figure illustrates why trajectory-level visualisation is useful as a first diagnostic layer: it allows rapid identification of the location discontinuities that may result from GPS glitches, spoofing-like behaviour or corrupted position data.

Selected telemetry variables and anomaly labels are shown in Figure 2. The figure presents the temporal behaviour of key flight and system parameters, including position-derived motion, altitude-related values, attitude changes, and battery-related signals. The injected anomaly intervals are marked against the corresponding signal changes, which makes it possible to visually compare abnormal events with their telemetry symptoms.

The normalised anomaly scores generated by the unsupervised models are compared in Figure 3. Although these methods are able to highlight several injected anomaly intervals, their score distributions also indicate false positive detections outside the labelled anomaly periods. This

confirms that unsupervised models are useful for ranking suspicious windows, but their outputs require correlation with rule-based indicators and domain interpretation.

The replay-like communication anomaly is illustrated in Figure 4. The figure focuses on the repeated command segment, where abnormal sequence behaviour and duplicated command patterns were introduced into the MAVLink-like stream. This visualisation supports the security-oriented part of the workflow by showing that communication-level anomalies may be detected not only through physical flight parameters, but also through message timing, repeated payloads, and sequence irregularities.

Listing 1 presents a simplified fragment of the proposed Python workflow. The complete runnable script accompanying this draft generates the synthetic dataset, calculates features, applies rule-based indicators, trains unsupervised models, as well as exports tables and figures.

Five-flight validation fixture

To make the experimental workflow reproducible alongside access-controlled public datasets, a five-flight MAVLink-like companion dataset was generated. This dataset is not presented as real flight evidence. It is an executable validation fixture that mirrors the structure expected from `tlog/.bin/.ulg` preprocessing. It contains one normal-like flight and four flights with controlled anomaly classes: GPS glitch/position jump,

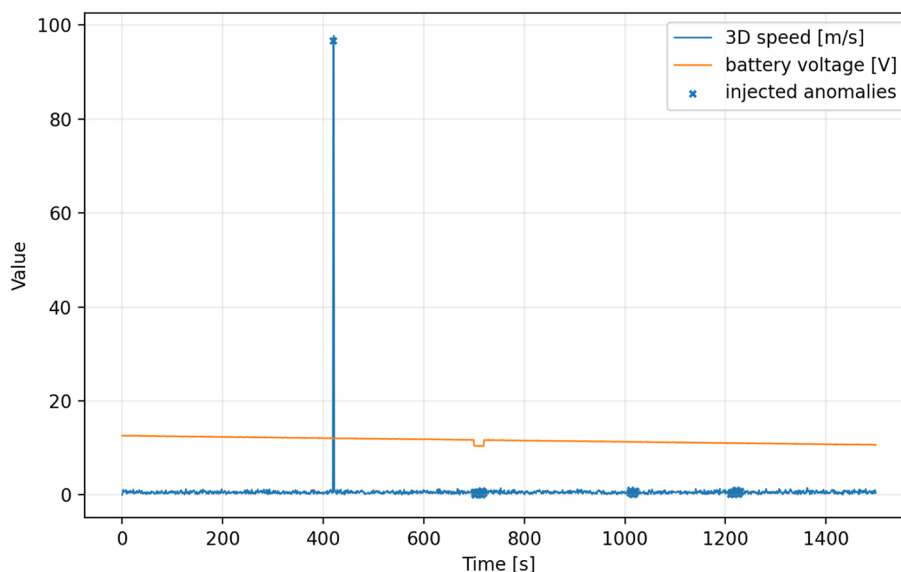


Figure 2. Selected telemetry signals and injected anomaly labels

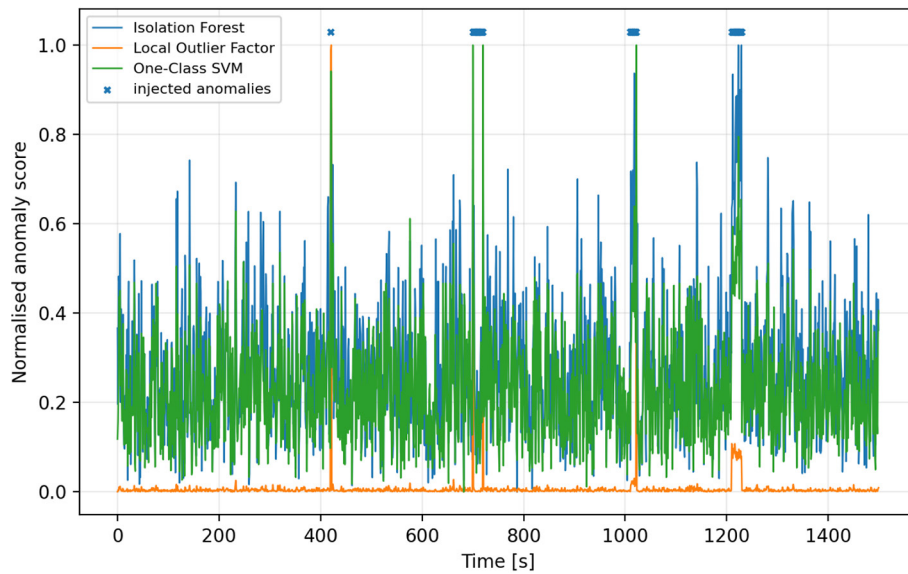


Figure 3. Normalised anomaly scores for unsupervised detection models

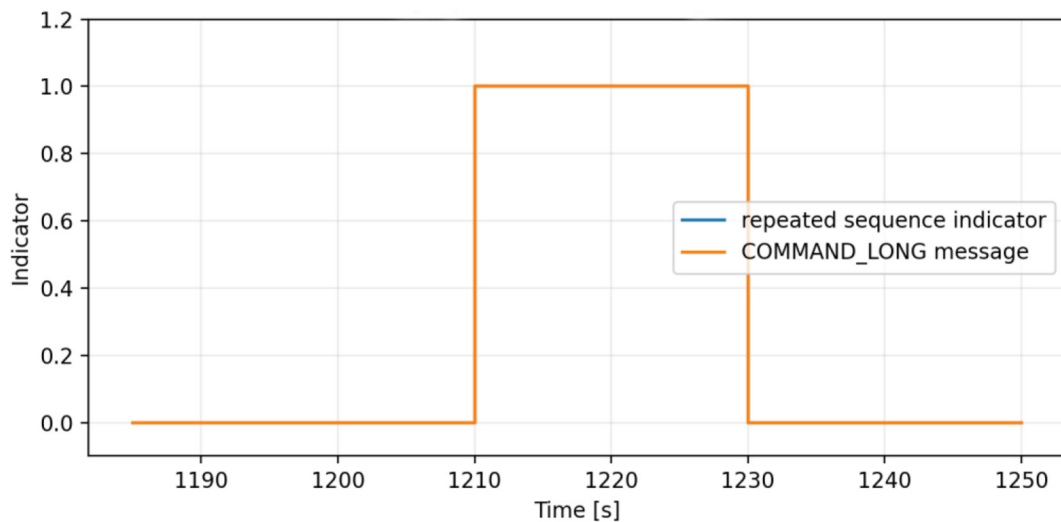


Figure 4. Replay-like command segment based on repeated sequence indicators

Listing 1. Python code for engineering and model-based anomaly detection

```
# Core feature engineering and model-based anomaly detection
feature_cols = [
    "horizontal_speed_mps", "vertical_speed_mps", "speed_3d_mps",
    "roll_rate_radps", "pitch_rate_radps", "yaw_rate_radps",
    "voltage_drop_vps", "current_delta_a", "repeated_seq", "duplicate_payload" ]
X = df[feature_cols].replace([np.inf, -np.inf], np.nan).fillna(0)
Xs = StandardScaler().fit_transform(X)

# Rule-based examples calibrated for the analysed platform and fixture
df["rule_gps_jump"] = (df["horizontal_speed_mps"] > 35).astype(int)
df["rule_voltage_sag"] = ((df["voltage_drop_vps"] > 0.35) |
    ((df["battery_v"] < 10.9) & (df["current_a"] > 15))).astype(int)
df["rule_replay_like"] = (((df["msg_type"] == "COMMAND_LONG") &
    (df["repeated_seq"] == 1)) | (df["duplicate_payload"] == 1)).astype(int)

# Unsupervised models
models = {
    "Isolation Forest": IsolationForest(n_estimators=200, contamination=contamination),
    "Local Outlier Factor": LocalOutlierFactor(n_neighbors=35,
        contamination=contamination),
    "One-Class SVM": OneClassSVM(kernel="rbf", gamma="scale", nu=contamination)
}
```

altitude-estimate divergence, battery/attitude disturbance, and replay-like command repetition. The fixture is used to verify feature engineering, detector behaviour, and result-export logic before the same schema is applied to real log files.

The generated five-flight dataset was used only as a controlled validation fixture, rather than the main empirical evidence of the study. Its purpose was to provide fully known labels for testing the behaviour of rule-based indicators and unsupervised models under repeatable conditions. The composition of this controlled benchmark is summarised in Table 9. This fixture complements the real-log validation by allowing precision, recall, F1-score, and ROC AUC to be calculated for predefined anomaly intervals. The corresponding detection performance of the proposed anomaly detection pipeline on the generated five-flight validation fixture is presented in Table 10. However, the results obtained from this fixture should not be generalised directly to real UAV operations without comparison against real annotated flight logs.

The extended validation confirms that anomaly reports should be interpreted at the event level, not only at the sample level. A GPS-glitch event should be interpreted through unrealistic horizontal displacement, GPS altitude inconsistency, and a matching STATUSTEXT or estimator warning. An altitude-estimate divergence should be interpreted through disagreement between GPS,

barometric, and canonical altitude estimates. A battery-related event should combine voltage-drop rate, current increase, and battery or failsafe messages. A replay-like event should combine repeated MAVLink command payloads, sequence irregularities, abnormal command timing, and command-state consistency.

Comparison of detector behaviour across evaluation settings

The difference between Tables 8 and 10 reflects the different role of the two controlled evaluation settings. Table 8 reports the outcome of a single initial simulation benchmark in which four anomaly types were injected into one controlled MAVLink-like trajectory. These anomalies were clearly separated in time and were intentionally designed to match the physical and communication-level assumptions of the rule-based indicators. Therefore, the rule-based layer achieved a high F1-score of 0.854.

In contrast, Table 10 reports the results for a more heterogeneous five-flight validation fixture. This fixture includes one normal-like flight and four flights with different anomaly mechanisms: GPS glitch, altitude/EKF divergence, power-and-attitude disturbance, and replay-like command repetition. The rule-based indicators maintained very high precision, but their recall decreased because not all labelled anomaly

Table 9. Five-flight validation dataset used to test the proposed anomaly pipeline

Flight	Samples	Duration (s)	Labelled anomaly points	Rule-detected points	Dominant anomaly	STATUSTEXT / event
F01	1200	1199	0	0	none	none
F02	1200	1199	20	21	GPS glitch	GPS glitch detected
F03	1200	1199	70	47	Altitude/EKF	EKF altitude variance
F04	1200	1199	60	2	Power and attitude	Battery failsafe; attitude divergence
F05	1200	1199	35	35	Replay-like command	Repeated command sequence

Table 10. Detection performance on the generated five-flight validation fixture

Method	Precision	Recall	F1-score	ROC AUC	Detected points
Rule-based indicators	0.981	0.557	0.710	0.778	105
Isolation forest	0.498	0.649	0.563	0.873	241
Local outlier factor	0.303	0.395	0.343	0.683	241
One-class SVM	0.502	0.654	0.568	0.867	241

intervals produced threshold-exceeding symptoms at every sample. This is particularly visible for the power-and-attitude disturbance, where the abnormal event is semantically meaningful at the flight-event level, but only a limited number of samples exceed the strict voltage or attitude-rate thresholds.

The behaviour of unsupervised models also changed across the two settings. Isolation Forest achieved a slightly higher F1-score in the five-flight fixture than in the initial simulation benchmark and, more importantly, a higher ROC AUC. This suggests that the method was more effective as a ranking mechanism for suspicious windows than as a binary classifier under a fixed threshold. Local Outlier Factor performed worse in the five-flight setting, most likely because the local density structure varied across flights and anomaly classes. One-Class SVM improved in the extended fixture, but still produced a considerable number of false positive detections. These results confirm that detector performance should be interpreted in relation to the evaluation setting, anomaly type, thresholding strategy, and event-level interpretation. A summary comparison of the two controlled evaluation settings is provided in Table 11.

Approximate uncertainty of controlled-fixture metrics

To improve transparency, Table 12 provides approximate 95% confidence intervals for the five-flight fixture. The intervals were reconstructed from the reported precision, recall, number of detected points, and labelled anomaly points. They should be interpreted cautiously because telemetry samples are temporally correlated and because the generated fixture is not a substitute for repeated real-flight evaluation. Nevertheless, the intervals show the relative uncertainty associated with the reported single-fixture metrics and support the conclusion that high precision of the rule-based layer is obtained at the cost of lower recall. The accompanying reproducibility files are: `generate_uav_validation_dataset.py`, `uav_generated_validation_5flights.csv`, `uav_generated_validation_summary.csv`, and `uav_generated_validation_model_results.csv`. The same processing schema can be used to parse real.tlog files with `pymavlink`, `ArduPilot.bin` files with `MAVExplorer/DFReader`, or `PX4.ulg` files with `pyulog`. For PX4 logs, topics such as `battery_status`, `vehicle_global_position`, `vehicle_gps_position`, `vehicle_local_position`, `vehicle_attitude`,

Table 11. Comparison of the two controlled evaluation settings

Aspect	Initial simulation benchmark	Five-flight validation fixture
Number of flights	1	5
Main purpose	Preliminary detector verification	Controlled multi-event validation
Data type	Synthetic MAVLink-like trajectory	Generated multi-flight fixture
Anomaly classes	GPS jump, voltage sag, attitude spike, replay-like segment	Normal flight, GPS glitch, altitude/EKF anomaly, power/attitude event, replay-like command
Labels	Fully controlled sample-level labels	Fully controlled sample- and event-level labels
Expected detector behaviour	Rules favoured by clearly defined anomalies	More heterogeneous behaviour across anomaly types
Main interpretation	Proof-of-concept benchmark	Reproducible controlled validation fixture

Table 12. Approximate uncertainty of five-flight validation metrics

Method	Precision 95% CI	Recall 95% CI	F1-score 95% CI	Interpretation
Rule-based indicators	0.933–0.997	0.485–0.626	0.655–0.761	High precision; conservative thresholds reduce recall.
Isolation forest	0.435–0.561	0.578–0.715	0.506–0.618	Good ranking ability, but many false positives under fixed threshold.
Local outlier factor	0.250–0.364	0.326–0.466	0.292–0.398	Weaker stability in heterogeneous multi-flight fixture.
One-class SVM	0.439–0.565	0.583–0.719	0.511–0.623	Similar to Isolation Forest, with comparable false-positive tendency.

vehicle_status, vehicle_command, and telemetry_status can be mapped to the feature schema used in this article.

INTERPRETATION

The simulated results should be interpreted from two perspectives: flight safety and communication security. Operational anomalies, such as GPS jumps, sudden altitude changes, attitude disturbances, or voltage drops, may result from sensor errors, environmental disturbances, aggressive manoeuvres, or logging artifacts. Communication anomalies, such as repeated commands or abnormal sequence behaviour, may indicate replay-like behaviour, but may also result from packet retransmission, parser effects, or unstable radio communication. This confirms the need for hybrid analysis and manual validation.

A specific limitation of MAVLink.tlog files is that they represent the ground-station perspective, rather than the complete onboard state of the UAV. Packet loss, delayed packets, retransmissions, parser artefacts, and irregular sampling may produce apparent discontinuities in telemetry. Such discontinuities may increase false positive detections, especially for the indicators based on inter-arrival time, sequence behaviour, or sudden changes between consecutive samples. Therefore, tlog-based anomaly detection should be treated as a screening layer. Whenever possible, suspicious windows should be cross-checked with onboard DataFlash or PX4 ULog data, system-status messages, and mission context. A major advantage of the proposed workflow is interpretability. Rule-based indicators provide clear explanations for domain experts, while unsupervised models can detect less obvious deviations in multivariate telemetry. However, rule-based thresholds require calibration for the specific platform, mission profile, and sensor configuration. The thresholds calibrated for a small UAV and a short mission cannot be transferred automatically to a different airframe, payload, battery configuration, or flight mode. This is why the revised workflow treats threshold selection as a calibration step, rather than as a fixed universal rule set.

Compared with deep learning approaches, lightweight unsupervised models may be easier to deploy and explain in practical UAV diagnostics. Autoencoder-based and graph-based models can capture richer nonlinear dependencies, but their

practical usefulness depends on calibration, availability of labelled or semi-labelled data, and compatibility with interpretable event-level analysis. In the present experiments, Isolation Forest and One-Class SVM provided stronger ranking ability in the five-flight fixture, but they also generated more false positive detections than the conservative rule-based layer.

Replay-like detection remains a limitation of the current study. The implemented indicator is appropriate for naive repeated-command patterns and duplicated payloads, but it should not be interpreted as a complete defence against replay attacks. Sequence-renumbered, timing-shifted, or partial replay scenarios may evade simple sequence-based rules. Therefore, future replay-oriented analysis should combine command-parameter similarity, acknowledgement behaviour, mission-context consistency, vehicle-state response, and cryptographic freshness mechanisms where available.

Although the present study focused on post-flight analysis, the workflow can be transferred to near-real-time monitoring. The preprocessing and feature-engineering steps can be implemented in sliding time windows, for example over the most recent 1–10 seconds of telemetry. Rule-based indicators are computationally lightweight and can be evaluated immediately after each new message or window. Unsupervised models, such as Isolation Forest or One-Class SVM, can provide anomaly scores for recent windows, although their thresholds should be calibrated carefully to avoid excessive false alarms during manoeuvres, packet loss, or temporary communication degradation.

A near-real-time implementation would require additional mechanisms, including buffering of asynchronous messages, robust handling of missing packets, adaptive thresholds for flight mode and mission phase, and alert aggregation to avoid repeated warnings for the same event. Therefore, real-time deployment is outside the scope of the current paper, but the proposed pipeline provides a practical basis for future online telemetry monitoring systems. A central limitation of the original experimental design was the strong role of generated data in the quantitative evaluation. In the revised interpretation, this limitation is addressed by making real-log validation the primary empirical stage and by treating the generated fixture only as a reproducibility benchmark. Nevertheless, the availability of fully annotated MAVLink telemetry logs remains limited.

Public datasets often provide either raw logs without detailed event-level labels or diagnostic outputs produced by external tools. Therefore, the present study should be interpreted as a feasibility-oriented validation of a hybrid MAVLink anomaly detection workflow, rather than as a complete operational benchmark across all UAV platforms, missions, and failure modes.

CONCLUSIONS

This paper proposed and evaluated a data analytics workflow for detecting anomalies in MAVLink telemetry logs of unmanned aerial vehicles. The approach integrates protocol-aware preprocessing, mathematical feature engineering, calibrated rule-based anomaly indicators, replay-like communication-pattern screening, public-log validation, and unsupervised anomaly detection models. The revised evaluation confirms that the proposed MAVLink anomaly detection workflow can be applied to real UAV logs and can identify suspicious windows consistent with diagnostic events, such as altitude-estimate divergence, attitude-control problems, EKF warnings, compass/variance flags, and selected communication irregularities. The generated five-flight fixture remains useful for reproducible quantitative benchmarking, but it should be interpreted as a controlled test environment, rather than as the main empirical basis of the study.

The main conclusion is that the MAVLink anomaly analysis should combine physical consistency checks, communication-level indicators and unsupervised scoring. This hybrid strategy improves the interpretability of suspicious telemetry windows and supports a more reliable distinction between operational anomalies, estimator-related warnings, power-related events and replay-like command patterns.

Future work should extend the evaluation to a larger set of independently annotated MAVLink, ArduPilot DataFlash, and PX4 ULog files, preferably including synchronised onboard and ground-station logs. Further research should also include repeated-run sensitivity analysis, platform-specific threshold calibration, more advanced replay-like and sequence-renumbered attack scenarios, and validation across different UAV platforms, mission profiles, and radio-link conditions.

REFERENCES

1. Dziuba-Koziel M., Koziel G., Markiewicz J., Kochanowicz M., Dorosz D., Miluski P. and Kisala P. (2025). Machine learning-assisted design of inverted gradient multiring optical fibres for flat-top beam generation. *Optics Express*, 34(4): 7159–7173, <https://doi.org/10.1364/OE.587143>
2. Dziuba-Koziel M., Koziel G., Harasim D., Kisala P. and Kochanowicz M. (2025). Method of automatic calibration and measurement of the light polarisation plane rotation with tilted fibre bragg gratings and discrete wavelet transform usage. *Advances in Science and Technology Research Journal*, 19(1): 165–177, <https://doi.org/10.12913/22998624/194890>
3. He Y., Peng Y., Wang S., Liu D. (2019). ADMOST: UAV flight data anomaly detection and mitigation via online subspace tracking. *IEEE Trans. Instrum. Meas.* 68(4), 1035–1044. <https://doi.org/10.1109/TIM.2018.2863499>
4. Yang L., Li S., Li C., Zhang A., Zhang X. (2023). A survey of unmanned aerial vehicle flight data anomaly detection: Technologies, applications, and future directions. *Sci. China Technol. Sci.* <https://doi.org/10.1007/s11431-022-2213-8>
5. Zhong J., Zhang Y., Wang J., Luo C., Miao Q. (2022). Unmanned aerial vehicle flight data anomaly detection and recovery prediction based on spatio-temporal correlation. *IEEE Trans. Reliab.* <https://doi.org/10.1109/TR.2021.3134369>
6. Yang L., Li S., Li C., Zhu C., Zhang A., Liang G. (2023). Data-driven unsupervised anomaly detection and recovery of unmanned aerial vehicle flight data based on spatiotemporal correlation. *Sci. China Technol. Sci.* <https://doi.org/10.1007/s11431-022-2312-8>
7. Yang L., Li S., Li C., Zhu C. (2024). Data-driven multivariate regression-based anomaly detection and recovery of unmanned aerial vehicle flight data. *J. Comput. Des. Eng.* <https://doi.org/10.1093/jcde/qwae023>
8. Yang L., Li S., Li C., Zhu C., Zhang A., Liao Z. (2024). Spatio-temporal correlation-based multiple regression for anomaly detection and recovery of UAV flight data. *Adv. Eng. Inform.* <https://doi.org/10.1016/j.aei.2024.102440>
9. Sezgin A., Keskin R., Boyaci A. (2025). Anomaly detection in unmanned aerial vehicle telemetry using automated machine learning. *Firat Univ. J. Eng. Sci.* <https://doi.org/10.35234/fumbd.1668498>
10. Bell V., Rengasamy D., Rothwell B., Figueredo G. (2022). Anomaly detection for unmanned aerial vehicle sensor data using a stacked recurrent autoencoder. *arXiv*. <https://doi.org/10.48550/arXiv.2203.04734>

11. Zhou S., He Z., Chen X., Chang W. (2024) An anomaly detection method for UAV based on wavelet decomposition and stacked denoising autoencoder. *Aerospace*. 11(5): 393. <https://doi.org/10.3390/aerospace11050393>
12. Chen H., Lyu Y., Shi J., Zhang W. (2024). UAV anomaly detection method based on convolutional autoencoder and support vector data description with 0/1 soft-margin loss. *Drones*. 8(10): 534. <https://doi.org/10.3390/drones8100534>
13. Feng C., Fan J., Liu Z., Jin G., Chen S. (2025). Unmanned aerial vehicle anomaly detection based on causality-enhanced graph neural networks. *Drones*. 9(6): 408. <https://doi.org/10.3390/drones9060408>
14. Feng C., Fan J., Jin G., Liu J., Chen S. (2025). STE-RS: A novel framework for robust anomaly detection in UAV flight data. *Results Eng.* <https://doi.org/10.1016/j.rineng.2025.106801>
15. Silalahi S., Ahmad T., Studiawan H. (2023). Transformer-based sentiment analysis for anomaly detection on drone forensic logs. In: 2023 11th International Symposium on Digital Forensics and Security (ISDFS); <https://doi.org/10.1109/ISDFS58141.2023.10131749>
16. Silalahi S., Ahmad T., Studiawan H., Anthi E., Williams L. (2024). Severity-oriented multiclass drone flight logs anomaly detection. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2024.3396926>.
17. Silalahi S., Ahmad T., Studiawan H., Anthi E., Williams L. (2025). Interpretable ordinal-aware with contrastive-enhanced anomaly severity detection on UAV flight log messages. *IEEE Access*. <https://doi.org/10.1109/ACCESS.2025.3580056>
18. Yang L., Li S., Zhang Y., Zhu C., Liao Z. (2024). Deep learning-assisted unmanned aerial vehicle flight data anomaly detection: A systematic review. *IEEE Sensors J.* <https://doi.org/10.1109/JSEN.2024.3451648>
19. Reis M.J.C.S., Reis A.J.D. (2025). Edge-based real-time fault detection in UAV systems via B-spline telemetry reconstruction. *Sensors*. 25(16): 4944. <https://doi.org/10.3390/s25164944>
20. ArduPilot Dev Team. Downloading and analyzing DataFlash logs. ArduPilot Documentation; 2024. Available from: <https://ardupilot.org/copter/docs/common-downloading-and-analyzing-data-logs-in-mission-planner.html>
21. DroneKit. DroneKit-LA test data. GitHub repository. Available from: <https://github.com/dronekit/dronekit-la-testdata>
22. Kabaoglu A., Sariel S. UAV-SEAD: State Estimation Anomaly Dataset for UAVs. Hugging Face; 2026. Available from: <https://huggingface.co/datasets/aykutkabaoglu/uav-flight-anomaly-dataset>
23. Lu Y. TLM:UAV anomaly detection datasets. Kaggle. Available from: <https://www.kaggle.com/datasets/luyucwnu/tlm-uav-anomaly-detection-datasets>