DroPTC: sentence-level drone flight log forensics using contrastive learning and explainable AI

Swardiantara Silalahi, Tohari Ahmad, Hudan Studiawan, Frank Breitinger

Angaben zur Veröffentlichung / Publication details:

Silalahi, Swardiantara, Tohari Ahmad, Hudan Studiawan, and Frank Breitinger. 2026. "DroPTC: sentence-level drone flight log forensics using contrastive learning and explainable AI." *Forensic Science International: Digital Investigation* 56 (Supplement): 302051. https://doi.org/10.1016/j.fsidi.2026.302051.

Forensic Science International: Digital Investigation 56 (2026) 302051

Contents lists available at ScienceDirect

Forensic Science International: Digital Investigation

journal homepage: www.elsevier.com/locate/fsidi

DFRWS EU 2026 - Selected Papers from the 13th Annual Digital Forensics Research Conference Europe

DroPTC: Sentence-level drone flight log forensics using contrastive learning and explainable AI

Swardiantara Silalahi [a], Tohari Ahmad [a], Hudan Studiawan [a,*], Frank Breitinger [b,1]

[a] Department of Informatics, Institut Teknologi Sepuluh Nopember, Surabaya, Indonesia
[b] Chair for Cybersecurity, Institute of Computer Science, University of Augsburg, Augsburg, Germany

ARTICLE INFO

Keywords:
Drone forensics
Root cause analysis
Semantic alignment
Interpretability
Contrastive learning
Information security
Infrastructure

ABSTRACT

Unmanned Aerial Vehicles (UAVs), commonly known as drones, are increasingly deployed across diverse application domains, raising critical challenges for digital forensic investigation following safety incidents and system failures. In drone investigations, systematic analysis of flight logs is essential for reconstructing events, identifying root causes, and supporting reliable incident attribution and risk mitigation. Because a message may contain multiple sentences, message-level analysis cannot precisely pinpoint which log segment indicates a problem. Therefore, this paper proposes DroPTC (**Dro**ne **P**roblem **T**ype **C**lassifier), an end-to-end framework to identify and classify problems at the sentence level. A rule-based segmenter is designed to segment log messages into sentences based on historical log characteristics. Using the resulting log sentences, a pre-trained embedding is fine-tuned using contrastive learning for semantic alignment. The integrated gradient is employed to enhance the model's interpretability, enabling admissible and trustworthy analysis. Sentence deduplication is utilized to identify unique log events, thereby reducing the analyst workload. Quantitative and qualitative analysis of the experimental results show that DroPTC outperforms the baselines in three aspects: performance, trustworthiness, and efficiency. This paper also presents a working open-source tool as the tested implementation of the proposed framework. The tool accepts the decrypted flight log file and produces a forensic report in HTML and PDF format.

1. Introduction

The number of drones and their use are increasing, primarily due to their wide range of industrial applications. Civilians use drones to capture aerial photographs and videos, either as a hobby or professionally (Alcántara et al., 2021). In the agricultural sector, drones are used to spray disinfectants on plants, thereby improving the efficiency of agricultural work (Josephat et al., 2025). A recent use case demonstrated that drones also deliver packages (Cicek et al., 2025). In the event of a disaster, drones can locate victims and determine their exact positions, or deliver first aid to individuals who cannot be reached (Quero and Martinez-Carranza, 2025). The military sector, on the other hand, utilizes drones for intelligence, surveillance, or even offensive attack (Ayamga et al., 2021). These diverse drone use cases contribute to growth in the consumer drone market, which is projected to reach 9.6 million deliveries in 2030 (Laricchia, 2022). The direct consequence of the increasing use of drones is a rise in incidents. To date, hundreds of cases have occurred worldwide.[2] To this end, several standards and regulations have been developed to ensure a safe environment for drone operations, followed by active engagement from the research community.

During a drone forensic analysis, investigators need to answer critical questions: *Were there problems during the flight? When did they occur? What are the probable root causes?* Attempting to answer these questions, the research community proposes solutions from runtime anomaly detection (Shar et al., 2022; Minn et al., 2024; Wang et al., 2024), malfunction detection (Editya et al., 2023), and anomaly severity detection (Silalahi et al., 2025). Based on these studies, the flight log is one of the most important artifacts to analyze due to its relevance to diverse incident or attack scenarios (Mantas and Patsakis, 2022). Therefore, an attempt has been made to use drone flight log messages to identify problems and estimate their severity simultaneously (Silalahi et al., 2025). Although it can help identify problems, it cannot determine which part of the drone under investigation is problematic. Moreover,

* Corresponding author.
E-mail addresses: hudan@its.ac.id (H. Studiawan), frank.breitinger@uni-a.de (F. Breitinger).
[1] URL: https://www.FBreitinger.de.
[2] https://drone-detection-system.com/drone-incidents/.

https://doi.org/10.1016/j.fsidi.2026.302051

Available online 24 March 2026

...
Weak GPS signal. Positional accuracy may be compromised. Please fly with caution.
Forward Obstacle Sensing is not functioning. Ambient Light is too weak.
Mobile device CPU fully loaded. Related performance will be affected.
Cannot Takeoff in Travel Mode. Exit Travel Mode.
Image transmission signal weak.; RC signal lost.
...

Fig. 1. Raw messages can have multiple sentences. Oftentimes, there is only one problem-indicating sentence in one log record. For a denser log, there may be more than one, as shown in Fig. 2.

Failsafe RTH.; Controller triggered aircraft to descend. Auto RTH canceled.; RC signal lost. Returning to home.; Image transmission signal weak. Adjust antennas and make sure they are perpendicular to flight direction of aircraft.; RC signal weak. Avoid blocking antennas and adjust antenna orientation.; RTH ascending.

Fig. 2. An example of a dense log that has 10 sentences. Out of them, there are only two problem-indicating logs, as highlighted in red. (For interpretation of the references to colour in this figure legend, the reader is referred to the Web version of this article.)

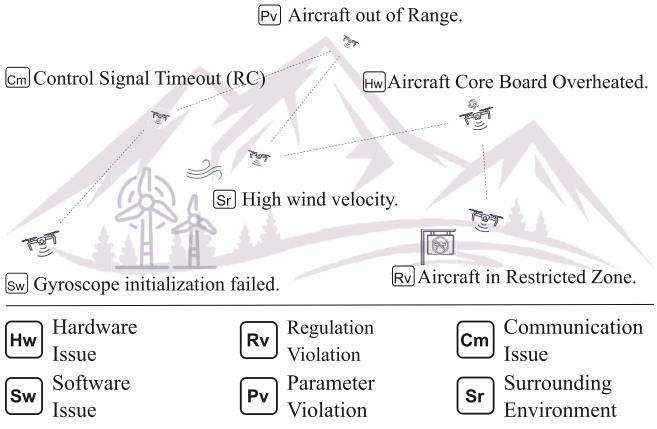

Fig. 3. An illustration of a drone flight where different types of problems occur.

the whole log record is classified into one class, while one message can contain multiple sentences, as shown in Fig. 1. Denser logs can contain up to ten sentences, as shown in Fig. 2. Furthermore, many neural-based methods operate as black boxes, hindering their admissibility as forensic evidence. Consequently, there is a need for interpretable, sentence-level classification that provides both fine-grained problem identification and transparent decision-making.

To address these gaps, this paper proposes DroPTC (**Dro**ne **P**roblem **T**ype **C**lassifier), an interpretable end-to-end framework to identify problems in drone flight logs at sentence-level granularity. Six problem types are introduced, i.e., hardware issue, software issue, communication issue, regulation violation, parameter violation, and surrounding environment, as illustrated in Fig. 3. By segmenting log messages into individual sentences and independently classifying each, DroPTC provides precise temporal and causal localization of problems that are critical for forensic investigations, while maintaining interpretability through integrated gradients (Sundararajan et al., 2017). DroPTC employs a rule-based segmenter to extract sentence-level log events and fine-tunes a pre-trained embedding using binary contrastive loss (Hadsell et al., 2006) to improve semantic alignment. To reduce analytical overhead, DroPTC deduplicates the events, enabling analysts to identify unique critical events and discard redundant entries. To promote reproducibility and facilitate adoption within the forensic community, we publicly release all code, trained models, and datasets.

Specifically, this paper attempts to answer the following research questions:

1. **RQ1**: How does DroPTC perform compared to existing baseline methods in terms of classification effectiveness, computational efficiency, and model trustworthiness?
2. **RQ2**: How do different embedding models affect DroPTC's classification performance, inference throughput, and trustworthiness metrics?
3. **RQ3**: How effective is the sentence-level deduplication in reducing analytical overhead by identifying unique problem-indicating events from drone flight logs?
4. **RQ4:** To what extent does DroPTC assist in identifying problem-indicating events in real-world flight logs?

The main *contributions* of this paper are as follows:

1. A sentence-level analysis paradigm for drone flight logs that reframes log records as sequences of discrete events to enable fine-grained temporal and causal problem localization. This paradigm opens new possibilities for pattern mining, causal analysis, and forensic reasoning in drone incident investigation.
2. An interpretable end-to-end system for sentence-level problem classification, featuring contrastive learning-based embedding fine-tuning, integrated gradient explanations, and sentence-level deduplication for analytical triage.
3. Publicly available datasets and comprehensive evaluation. Training dataset, case study logs from controlled problem-inducing scenarios with known ground truth, and systematic evaluation of the effectiveness, efficiency, and trustworthiness aspects.
4. An open-source implementation along with a complete artifact including model training, inference, and forensic workflow to facilitate adoption, reproducibility, and future research.

The remainder of the paper is organized as follows. Section 2 reviews related work. Section 3 details the proposed framework. Section 4 describes the experimental design and settings. Section 5 presents the experimental results and analysis. Section 6 concludes the paper and discusses future research directions.

2. Background and related work

This section discusses the literature about research on drones (Sec. 2.1) and the application of xAI in cybersecurity and forensics (Sec. 2.2).

2.1. Research on drones

The rapid increase in drone usage has made reliability and safety critical emerging research topics. Consequently, a substantial body of work has focused on fault and anomaly identification, detection, and localization (Puchalski and Giernacki, 2022). Generally, existing work on these topics can be categorized into three methodological approaches: model-based, signal processing-based, and data-driven. These paradigms primarily focus on numerical sensor data. However, each sensor data point analyzed can only support one conclusion. Moreover, analyzing sensor data individually can miss contextual dependencies among sensors and functionalities.

Combining several columns from a drone flight dataset into a feature vector for analysis can capture relationships among the variables. For instance, a long short-term memory (LSTM)-based anomaly detection model is trained on a feature vector comprising flight status and state units (Shar et al., 2022). Flight status includes flight mode and gain, while state units denote sensor readings of acceleration at the X-axis, Y-axis, and Z-axis. Given the most recent flight data, the model predicts the next value of state units. If the predicted state unit exceeds the

predefined threshold, then it is flagged as an anomaly. Correlation analysis is used to confirm the predicted anomaly (Shar et al., 2022). However, relying on normal data to train the model for anomaly detection is prone to overfitting. Moreover, the LSTM model's reasoning behind a prediction is not accessible to the analyst. Therefore, a solution proposed by Tan et al. (2025) combines rule-based and unsupervised learning to build a more generalizable and interpretable detection model.

Existing work on runtime anomaly detection has focused on identifying deviations in sensor data readings online. In a forensic setting, artifacts are analyzed post-incident, with emphasis on the scope and admissibility of the analysis. Following the trend in drone adoption, drone forensics has emerged as a subfield of digital forensics. The early stage of drone forensics research is characterized by case study papers that present reports on specific drone models after a thorough analysis of both physical and digital evidence (Studiawan et al., 2023). Over time, new brands and manufacturers have been established, complicating the landscape of digital forensics solutions for drone devices. Specifically, the encrypted telemetry and flight logs produced by DJI-made drones make it challenging to investigate without proprietary tools. Despite that, DROP (Clark et al., 2017) and GRYPHON (Mantas and Patsakis, 2019) emerged as the early open-source drone forensic tools that can parse and extract relevant information from telemetry and flight logs.

Similar to other types of aerial vehicles, drones record many events during operation that span the entire drone ecosystem, from the drone itself to the controller and the surrounding environment. As a result, flight logs constitute the most critical evidence in forensic investigations (Mantas and Patsakis, 2022). Out of hundreds of columns in a flight log file, some columns contain human-readable messages (Silalahi et al., 2023, 2025). Unlike raw telemetry or sensor data, log messages explicitly describe the contextual interactions among drone components and functionalities. They provide interpretable representations of the underlying system behavior. Each message reflects coordination among subsystems, including navigation, communication, propulsion, and control. Thereby, it reveals the operational dynamics of the entire drone ecosystem. Consequently, analyzing these log messages enables investigators to infer the causes and effects of flight events and system states in a holistic view, offering insights beyond isolated sensor readings or component-level anomalies (Silalahi et al., 2025).

Based on the literature review, no studies have used human-readable messages in flight logs at the sentence level for root cause analysis. Therefore, this paper proposes an interpretable framework to assist in investigating and finding the probable root cause of an incident.

2.2. xAI in cyber security and forensics

The use of artificial intelligence (AI) and machine learning (ML) has become essential in developing modern cybersecurity solutions, from malware analysis to network intrusion detection (Zhang et al., 2022). However, the most powerful of these models, particularly those based on deep learning, often function as “black boxes” (Zhang et al., 2022). This lack of transparency poses a significant challenge to adoption in security domains, as entrusting critical decisions to a system that cannot justify its reasoning is inherently risky (Capuano et al., 2022). This challenge has triggered the rapid growth of explainable AI (xAI), a field dedicated to making AI decisions and actions understandable to human users and stakeholders (Rjoub et al., 2023). The goal of xAI is to ensure accountability, thus balancing the trade-off between the advanced capabilities of AI and the practical needs of security professionals and forensic investigators.

While operational cybersecurity focuses on real-time defense, digital forensics focuses on post-incident investigation of cybercrimes or incidents to collect and analyze evidence for legal process (Alam and Altiparmak, 2024). Digital forensics must follow strict standards for AI systems, leading to the emergence of explainable AI for cyber forensics (Hall et al., 2022; Hargreaves et al., 2024). In a forensic context, an AI tool's output must be trustworthy, interpretable, understandable, and interactive to meet legal and ethical standards (Alam and Altiparmak, 2024). A notable example of xAI for digital forensics includes MetaCluster, which can provide explainable predictions on various modalities such as image, text, and statistical features. Another example applies xAI in deepfake detector models named DDL (Deepfake Detector Lens), which can provide decent feature attributions from image, frequency domain, and video. DDL successfully improved the interpretability of deep learning-based deepfake detectors, leading to transparent, reliable, and trustworthy analysis (Sun et al., 2025).

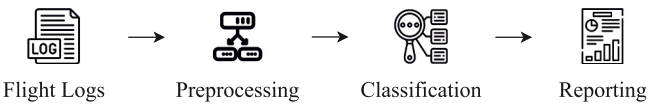

Fig. 4. The high-level flow of the proposed problem identification framework.

Following the standard practice in xAI for digital forensics, this paper employs an integrated gradient to investigate the relevant words in the input logs towards the predicted label. It provides the analyst with reasons why the model assigns a certain label to an input log. Therefore, the analysis produced by DroPTC can be accountable and trusted.

3. Proposed approach: DroPTC

Fig. 4 presents the workflow of DroPTC. This section describes the system's internal mechanisms in detail. The first subsection outlines the construction of the forensic timeline and the segmentation of messages. Following segmentation, the second subsection explains semantic feature extraction and classification, including the handling of class imbalance. Finally, the third subsection details the reporting process.

3.1. Pre-processing

Forensic timeline construction. DroPTC accepts the decrypted flight logs extracted from the smartphone used to control the drone. From the flight logs, human-readable messages are extracted from the `APP.tip` and `APP.warning` columns, along with the time stamps from the `CUSTOM.date [local]` and `CUSTOM.updateTime [local]` columns, to construct the forensic timeline. The messages generated during a flight are triggered by various events and conditions. Many of these messages contain multiple sentences, and each sentence implies distinct semantics. For this reason, message-level analysis for problem identification lacks precision and granularity. As shown in Fig. 5, even if the message-level samples are annotated with multiple labels (multi-label classification), one cannot determine which label belongs to which segment within the input message. Moreover, if there is more than one sentence with the same label present in the same input sample, the multi-label paradigm cannot preserve the frequency of the labels per input sample, since the label is converted into a multi-hot vector. Therefore, sentence-level analysis is proposed to preserve label frequency and improve forensic utility by assigning labels to each sentence, with a preprocessing step that segments messages into sentences.

Message segmentation. Based on the collection of drone flight log messages acquired from AirData UAV (2023), a rule is designed to slice input messages into sentences. A full stop is used as the first segment separator; decimal numbers are excluded. Then, a comma is used as the second segment separator while excluding large numbers (as in currency). Lastly, some messages use a colon as a segment separator. We exclude cases where a colon is not a sentence separator, as in “`Remote ID functionality normal (Code: 1B080003)`.” To preserve the original structure of the log messages, an ID is assigned to each message before segmenting them. The message ID is then used to reconstruct the input message back to its original structure.

Timestamp	Raw Message	Sentence-level Label	Message-level Label
...	...		
5/12/2025 8:21:05.74	RC signal lost.; Aircraft braking.; RC signal lost. Aircraft will follow preset action for lost signal.; Image transmission signal weak. Adjust antennas and make sure they are perpendicular to flight direction of aircraft.	[Communication Issue, Normal, Communication Issue, Normal, Communication Issue, Normal]	[Communication Issue, Normal]
...	...		

Fig. 5. An input sample contains 6 sentences with Normal and Communication Issue labels present. The sentence-level paradigm assigns a label to each sentence, preserving the label frequency.

3.2. Sentence-level classification

Model architecture. Identifying problems within drone flight logs constitutes a text-classification task whose objective is to build a model capable of distinguishing problem-indicating sentences from regular ones. Since one log record can contain multiple sentences, pinpointing expressions that indicate problems at message-level input is challenging, as illustrated in Fig. 1. For this reason, this paper proposes an end-to-end drone problem identification and classification framework to detect and categorize problem-indicating log segments, instead of log records. Analyzing logs at the segment level aims to precisely pinpoint problem-expressing log segments within a log record, to improve the utility of the analysis. Having each segment labeled, one can further analyze the relationship among the log segments regarding the occurring event.

DroPTC consists of three steps: pre-processing, classification, and reporting. As illustrated in Fig. 4, the log messages acquired from drone flight logs, $\mathbf{M} = [\mathbf{m}_1, \mathbf{m}_2, \ldots, \mathbf{m}_{|\mathbf{M}|}]$, are segmented into sentences, $\mathbf{m} = [\mathbf{s}_1, \mathbf{s}_2, \ldots, \mathbf{s}_{|\mathbf{m}|}]$. After segmentation, each sentence is treated independently as one classification instance, which is later fed to a pre-trained embedding model, E_θ, to obtain the word vectors, $\mathbf{X} \in \mathbb{R}^{L \times d_{model}}$, with L and d_{model} denoting the maximum sequence length and the dimension of the vector based on the embedding model used. Mean pooling is employed to get the final sequence embedding by excluding the padding tokens, which is given by $\mathbf{h} = (\sum_{i=1}^{L} m_i \mathbf{X}_i)/(\sum_{i=1}^{L} m_i)$, where $m_i = \{0, 1\}$ denotes the attention mask with 1 indicating non-padding tokens, and $\mathbf{X}_i$ is the i-th row of $\mathbf{X}$. Then, the hidden state is fed to a linear layer followed by an activation function, $\mathbf{z}_1 = \mathrm{ReLU}(W_1^\top \mathbf{h} + \mathbf{b}_1)$, a dropout layer, $\mathbf{z}_2 = \mathrm{Dropout}(\mathbf{z}_1, p)$, and an output layer, $\mathbf{o} = W_2^\top \mathbf{z}_2 + \mathbf{b}_2$, where $W_1 \in \mathbb{R}^{d_{hidden} \times d_{model}}$ and $\mathbf{b}_1 \in \mathbb{R}^{d_{hidden}}$ are the parameter of the first linear layer, $W_2 \in \mathbb{R}^{d_{hidden} \times |C|}$ and $\mathbf{b}_2 \in \mathbb{R}^{|C|}$ are the parameter of the output layer, and p is the dropout ratio. Finally, a softmax is employed to get the predicted class probabilities, $\hat{\mathbf{y}} = \mathrm{softmax}(\mathbf{o}) \in \mathbb{R}^{|C|}$ where $|C| = 7$ denotes the number of target classes. The whole network is trained using standard cross-entropy as the loss function.

Class Imbalance handling. Given the class imbalance in the dataset, class weights are incorporated into the loss function to penalize minority-class predictions more heavily than majority-class predictions. Balanced and inverse class weights are explored, which can be computed by the following equations:

$$\boldsymbol{\alpha}_c^{\mathrm{bal}} = \frac{|\mathbf{M}|}{|C| \cdot |\mathbf{M}_c|} \tag{1}$$

$$\boldsymbol{\alpha}_c^{\mathrm{inv}} = \left(\frac{|\mathbf{M}_c|}{|\mathbf{M}|}\right)^{-1} \tag{2}$$

Semantic alignment via contrastive fine-tuning. Text embedding has been rapidly developing over the past decade, enabling the extraction of features from text. Several general-purpose pre-trained embeddings have recently become available, enabling a wide range of natural language processing (NLP) tasks. Many of these models are trained on a large, general-purpose corpus using self-supervised techniques to capture the corpus's semantics. To make the most of it, one can fine-tune the pre-trained model on a domain-specific dataset to better capture the domain's semantics. Thus, the model perceives the input text as being more closely aligned with human expert judgment (Silalahi et al., 2025). In a forensic context, it is critical to ensure the neural model's trustworthiness and transparency for an admissible investigation. Such models were once black boxes and are far less transparent than they are today.

Table 1
The distribution of the problem types in the model development dataset.

Problem Type	Train	Test	Total
Normal	521	132	653
Hardware Issue	114	28	142
Software Issue	87	20	107
Surrounding Environment	51	13	64
Parameter Violation	41	10	51
Regulation Violation	11	3	14
Communication Issue	14	3	17
Total	839	209	1048

To this end, pre-trained sentence embeddings, all-MiniLM-L6-v2 and all-mpnet-base-v2, are explored in this paper. Binary contrastive loss (Hadsell et al., 2006) is used to understand the semantic landscape of logs tailored to the drone problem type classification task. The objective of this process is to improve the model's understanding of the drone log semantics in pinpointing problem-expressing log segments. The integrated gradient (Sundararajan et al., 2017) is employed to investigate which words are considered relevant by the model when predicting an input log segment. Consequently, one can check and ensure that a deep neural model behaves in a way that can be trusted to perform evidence analysis and produce admissible analysis.

3.3. Reporting

Event deduplication for data triage. The challenge is to develop a model that can automatically identify problem-expressing logs and classify them into predefined classes. Following the procedure explained in Sec. 3.2, each input log record is segmented into sentences, and each sentence is then categorized into seven labels. After that, a sentence deduplication is employed to group syntactically similar log segments to obtain a brief overview of how many unique log segments exist, as well as the frequency distribution. This is needed for data triage during investigation to foster the evidence analysis process.

Report Generation. Finally, a forensic timeline along with problem type labels for each log segment is constructed as an attachment to the forensic report for each investigated drone flight log file. Other than that, a summary consisting of event frequency in a Gantt-style chart is constructed for the analyst to pinpoint when a certain problem occurred during the flight.

4. Experimental design

This section details the experimental design, comprising dataset preparation, baseline methods for comparison, and experimental settings related to the hardware used to run the experiment, evaluation metrics, and hyperparameter settings.

Table 2
The proposed problem type label is complying with well-defined standards (Sun and Hubbard, 2025) and existing work (Shar et al., 2022).

Proposed Category	NTSB Category	Illustrative Example
Hardware Issue	Machine/ Material	Motor/gimbal stuck, Propeller improperly installed
Software Issue	Machine/ Material	Sensor activation or initialization failure, changing flight mode failure
Regulation Violation	Human	Flying over people, flying beyond visual line of sight (BVLOS)
Parameter Violation	Human	Exceeding speed/altitude limits, ignoring system warnings
Communication Issue	Machine/ Human	RC signal lost, image transmission signal lost
Surrounding Environment	Environment	High wind velocity, magnetic interference, ambient light too low

Table 3
The problem-indicating events along with the problem types within each flight log to conduct a real-world case study.

No.	Problem-indicating Events
1	[Pv] Gimbal pitch axis endpoint reached. [Cm] Image transmission signal weak. [Cm] RC signal lost.
2	[Cm] Image transmission signal weak. [Cm] RC signal lost. [Cm] No image transmission. [Cm] RC signal weak. [Pv] Gimbal pitch axis endpoint reached.
3	[Cm] Image transmission signal weak. [Cm] RC signal lost. [Cm] No image transmission. [Cm] RC signal weak. [Cm] Downlink Lost.
4	[Sw] Gimbal auto check failed. [Hw] Gimbal stuck. [Cm] Image transmission signal weak. [Cm] RC signal lost. [Cm] Downlink Lost. [Cm] RC signal weak.
5	[Hw] Compass error. [Sr] Ambient light too low. [Cm] RC signal lost. [Pv] Gimbal pitch axis endpoint reached. [Sr] Downward ambient light too low. [Cm] RC signal weak.
6	[Hw] Compass error. [Sr] Downward ambient light too low. [Sr] Forward ambient light too low. [Hw] Low battery. [Hw] Remaining battery only enough for RTH. [Hw] Low battery RTH.
7	[Sr] Forward ambient light too low. [Sr] Downward ambient light too low. [Sr] Ambient light too low. [Cm] RC signal lost. [Cm] Image transmission signal weak. [Cm] Image transmission signal lost. [Cm] Downlink Lost. [Cm] RC signal weak.

4.1. Dataset preparation

Model development. The messages used in this experiment are obtained from AirData (AirData UAV, 2023) and Silalahi et al. (2023). Unique messages are merged, manually labeled, and split into train and test with an 80:20 ratio. Table 1 shows the class distribution after splitting. The dataset covers a wide range of drone models from different brands, as explained in the respective sources. The proposed problem type label is aligned with a well-defined standard, as shown in Table 2. Therefore, DroPTC can assist in a forensic investigation to find the probable root cause of an incident by pinpointing the problem-expressing log segment.

Case study. A DJI FPV[3] is used to produce seven flight logs, which are then analyzed in a case study. The first three flight logs emulate the communication issue problem by flying the drone through a building, blocking the communication between the remote controller and the drone itself. The next three flight logs emulate the surrounding environment issue by flying the drone in the evening. The case study dataset covers five out of six problem types, as summarized in Table 3. The process of constructing model development and case study dataset follows the standard practice and the FAIR (findable, accessible, interoperable, reusable) principle in the creation of digital forensics dataset (Mombelli et al., 2024).

[3] https://www.dji.com/id/downloads/products/dji-fpv, Accessed October 2025.

Table 4
The base model used as the pre-trained embeddings.

Embedding	d_{model}	Parameter
all-MiniLM-L6-v2	384	22.7M
all-MPNet-base-v2	768	109M
BERT-base-uncased	768	109M
ModernBERT-base	768	150M
NeoBERT	768	245M

4.2. Baselines

Several relevant baselines are chosen for performance comparison, including drone anomaly detection based on human-readable logs (Silalahi et al., 2024, 2025) and transformer-based system log anomaly detection (Le and Zhang, 2021; Pham and Lee, 2023). The architecture of these baselines is implemented based on their best-performing configurations. Apart from them, we construct baselines by varying the embedding models used to support DroPTC. The chosen embeddings include the standard word embedding model, BERT-base-uncased (Devlin et al., 2019), standard sentence embedding model (Reimers and Gurevych, 2019), such as all-MiniLM-L6-v2 and all-mpnet-base-v2, and recent models such as ModernBERT-base (Warner et al., 2025) and NeoBERT (Breton et al., 2025). Table 4 shows the comparison between the embedding models' parameters and dimensional size.

4.3. Experiment settings

The experiment is conducted on a computer with an Ubuntu operating system, an i9-12900K CPU, and a NVIDIA GeForce RTX 3080 Ti (12 GB) GPU. The model is trained using the AdamW optimizer with a learning rate of $2e-5$, 20 epochs, and a batch size of 8. Hyperparameters are adopted from Silalahi et al. (2025). Model checkpoints are saved at each epoch, and the checkpoint with the highest F1-score on the test set is selected for final evaluation. To ensure robustness, each configuration is trained 10 times with different random seeds. Given the forensic context, we evaluate DroPTC across three dimensions:

Effectiveness. We report the mean and standard deviation of the weighted-average precision, recall, and F1-score, along with accuracy, across 10 runs. Paired t-tests and Wilcoxon signed-rank tests are used to assess the significance of performance gains across model configurations.

Efficiency. We measure inference throughput (samples/second) in both CPU-only and GPU modes across varying sample sizes (100, 250, 500, 750, 1000, 2500, 5000, 7500, and 10,000 samples). Tests are conducted to assess computational feasibility and scalability on varying log volumes for different deployment scenarios. A warm-up is used to eliminate the effect of model loading time for the first time.

Trustworthiness. We evaluate three aspects of model reliability: calibration through expected calibration error (ECE), confidence distribution comparing correct vs. incorrect predictions via violin plots, and prediction stability across the 10 runs using three metrics: cross-run stability by taking the mean (over all sample) of per sample standard deviation (over ten runs) of the prediction confidence, number of samples with high variance (std > 0.1) indicating unstable predictions, and average prediction diversity (where 1 indicates perfect consistency). Additionally, we examine integrated gradient attributions on 77 abnormal samples to investigate the semantic alignment of the fine-tuned embeddings. An expert manually extracts the highly relevant words from each sample. Then, the macro-average precision of the per-sample top-k words, computed using the attribution score, is reported.

5. Experimental results and analyses

After experimenting with the proposed framework, this section presents the results in detail.

Table 5
Performance ($mean_{std}$) comparison against baselines.

Model	Accuracy	Precision	Recall	F1
DroneLog	$0.911_{0.009}$	$0.913_{0.012}$	$0.911_{0.009}$	$0.909_{0.011}$
DroLoVe	$0.915_{0.007}$	$0.919_{0.008}$	$0.915_{0.007}$	$0.915_{0.007}$
NeuralLog	$0.917_{0.008}$	$0.922_{0.009}$	$0.917_{0.008}$	$0.918_{0.008}$
TransSentLog	$0.919_{0.011}$	$0.922_{0.009}$	$0.919_{0.011}$	$0.918_{0.010}$
DroPTC	$\mathbf{0.932_{0.006}}$	$\mathbf{0.933_{0.006}}$	$\mathbf{0.932_{0.006}}$	$\mathbf{0.930_{0.005}}$

Table 6
Effect of different pre-trained embeddings on DroPTC's performance ($mean_{std}$).

Base Model	Accuracy	Precision	Recall	F1
ModernBERT	$0.908_{0.012}$	$0.909_{0.012}$	$0.908_{0.012}$	$0.906_{0.012}$
NeoBERT	$0.913_{0.010}$	$0.917_{0.010}$	$0.913_{0.010}$	$0.911_{0.011}$
BERT-base	$0.918_{0.007}$	$0.921_{0.006}$	$0.918_{0.007}$	$0.917_{0.006}$
MiniLM-L6	$0.921_{0.008}$	$0.929_{0.011}$	$0.921_{0.008}$	$0.922_{0.009}$
MPNet-base	$\mathbf{0.927_{0.007}}$	$\mathbf{0.931_{0.006}}$	$\mathbf{0.927_{0.007}}$	$\mathbf{0.927_{0.006}}$

Table 7
Ablation study on the DroPTC's components toward the performance ($mean_{std}$).

Model Variant	Accuracy	Precision	Recall	F1
DroPTC	$\mathbf{0.932_{0.006}}$	$\mathbf{0.933_{0.006}}$	$\mathbf{0.932_{0.006}}$	$\mathbf{0.930_{0.005}}$
w/o class weight	$0.929_{0.007}$	$0.931_{0.008}$	$0.929_{0.007}$	$0.928_{0.007}$
w/o fine-tuning	$0.924_{0.007}$	$0.928_{0.007}$	$0.924_{0.007}$	$0.924_{0.006}$
w/o both	$0.927_{0.007}$	$0.931_{0.006}$	$0.927_{0.007}$	$0.927_{0.006}$

5.1. Performance evaluation and comparison

Effectiveness. Unlike prior work that focuses solely on accuracy, our evaluation ensures DroPTC meets forensic admissibility requirements. The main result of the model development experiment is shown in Table 5. Overall, DroPTC, with the fine-tuned MPNet and inverse class weight, outperforms all four baselines from the existing works related to log problem identification and anomaly detection. DroPTC achieves an F1 score of 0.930 ± 0.005, higher than TransSentLog as the best-performing baseline, with an F1 score of 0.918 ± 0.01. To verify the performance improvement, a series of statistical tests is conducted between the best-performing DroPTC scenario and TransSentLog. The paired *t*-test confirmed the difference is statistically significant, with a p-value of 0.0113, which translates to medium-to-large practical improvement, based on Cohen's *d* of 1.003 effect size. The Wilcoxon signed-rank test showed a similar trend, with a *p*-value of 0.0152.

Given the relatively small dataset size, which is one reason to use pre-trained embeddings, an experiment is conducted to investigate the impact of various embedding models listed in Table 4 on DroPTC's performance. As shown in Table 6, performance with a lightweight model, such as all-MiniLM-L6-v2, is comparable to that of a larger model, such as all-mpnet-base-v2. A *t*-test comparing fine-tuned MPNet and MiniLM shows no statistically significant difference, with a p-value of 0.2924. Interestingly, even without fine-tuning, DroPTC's performance with a small-size embedding is better than that of mainstream embeddings, such as BERT, and modern models, including NeoBERT and ModernBERT. A t-test comparing pre-trained MiniLM and BERT-base shows a statistically significant performance improvement ($p = 0.0353$). Table 7 further confirms that the performance gain is obtained from employing contrastive fine-tuned embedding and incorporating class weight into the loss function during training.

Efficiency. An efficiency test is conducted on the CPU to investigate the feasibility of running DroPTC on devices with limited computational resources. The test measures the processing time needed to perform batched prediction with a batch size of 32, to analyze the effect of employing different embedding models. The pre-processing and reporting steps are excluded to isolate the test exclusively to the model's prediction process. Fig. 6 shows the runtime of DroPTC's inference pipeline on varying log volumes. It can be seen that DroPTC with a small-size embedding can process nearly 1000 samples/second on CPU, which is 5x faster than the best-performing baseline. The best-performing DroPTC with a bigger embedding can process 144.87 samples/second on average, which is still acceptable given that the number of log sentences in a flight log of a single flight is unlikely to exceed 10,000. Consequently, it is feasible to deploy DroPTC on an analyst workstation that is not equipped with a GPU. The trade-off between performance and efficiency is presented in Fig. 6.

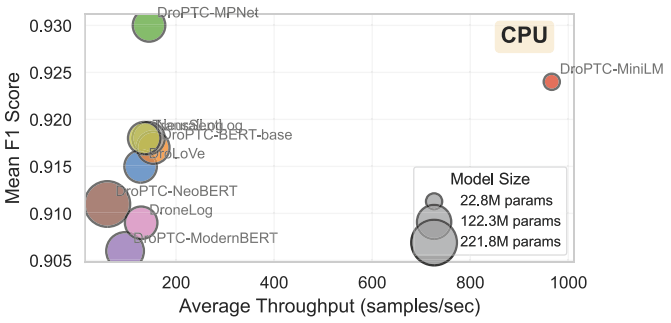

Fig. 6. The trade-off between performance and efficiency. DroPTC with small-size embedding is significantly more efficient compared to baselines.

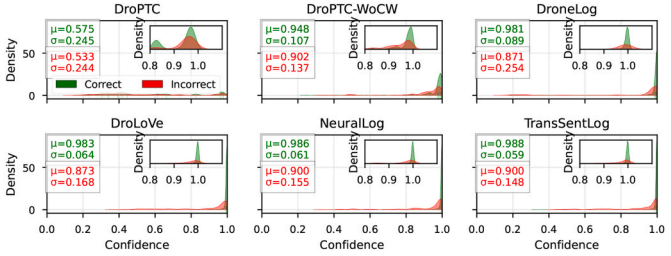

Fig. 7. The confidence distribution of correct and incorrect predictions.

Table 8
The effect of embedding fine-tuning and classifier training strategy on the DroPTC's prediction confidence.

Comparison	Δ Mean Conf.	Cohen's d	p-value
Pre-trained vs. Fine-tuned embeddings			
w/Frozen parameters	+0.2980	80.221	<0.001
w/Full fine-tuning	+0.0062	−0.139	0.6703
Frozen parameters vs. Full fine-tuning			
on pre-trained embedding	+0.3390	55.374	<0.001
on fine-tuned embedding	+0.0348	0.806	0.0313

Trustworthiness. Beyond effectiveness and efficiency, we assess the trustworthiness of the model's predictions through an analysis of prediction confidence. Fig. 7 shows the distribution of the prediction

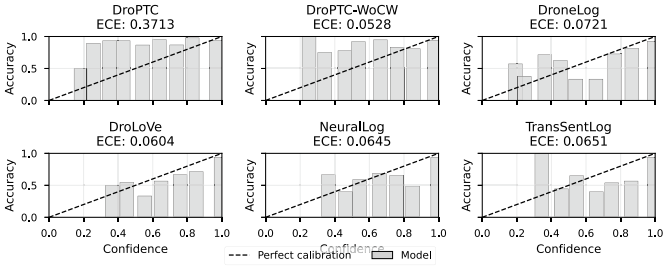

Fig. 8. The reliability diagram of DroPTC compared to baselines.

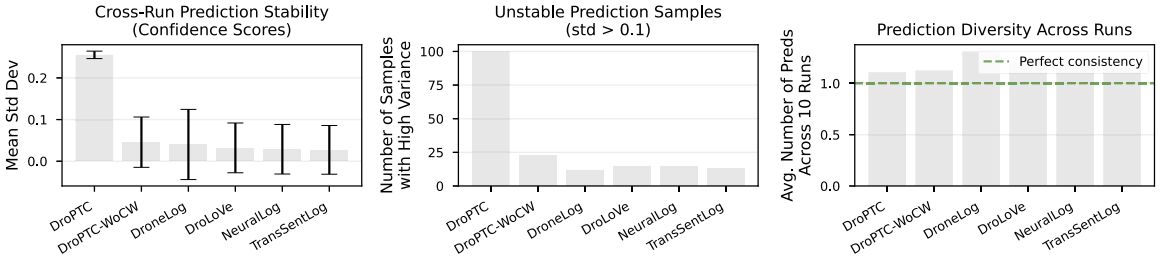

Fig. 9. The stability of per-sample prediction across 10 runs.

Table 9
The macro average precision of the most relevant words based on the attribution scores.

Model	Precision@k		
	$k = 1$	$k = 2$	$k = 3$
DroLoVe	0.909	0.838	0.814
DroneLog	0.818	0.799	0.801
DroPTC	0.805	0.831	0.788
DroPTC-WoCW	**0.935**	**0.903**	**0.827**
NeuralLog	0.883	0.857	0.814
TransSentLog	0.922	0.864	0.810

confidence over all samples in the test set, both for correct and incorrect predictions. It can be seen that DroPTC with the class weight is the least confident in both correct and incorrect predictions. Without the class weight, DroPTC-WoCW exhibits better confidence. This is due to the fine-tuned embedding, which exposes DroPTC to a domain-specific corpus, thereby improving its understanding of the semantics of flight log messages. Table 8 confirms that confidence gains are obtained from the embedding fine-tuning stage.

Further results on the reliability of DroPTC, in terms of the calibration between confidence score and performance gains, are shown in Fig. 8. Based on the figure, DroPTC yields an underconfidence verdict due to the class weight. This means that lower confidence is not associated with lower accuracy. On the other hand, DroPTC-WoCW achieves the best calibration, confirming that higher confidence scores are accompanied by higher accuracy. Additionally, DroPTC-WoCW exhibits stable performance as a result of fine-tuned embedding, supported by the stability analysis presented in Fig. 9. DroPTC-WoCW has the lowest mean standard deviation of predicted probabilities across all test samples. The second bar graph indicates that DroPTC-WoCW has the fewest samples with high variance. Nevertheless, in terms of prediction consistency across 10 runs, DroPTC shows the most consistent results among the baselines, as shown in the third graph.

In addition to predicting confidence, the integrated gradient is used to assess the importance of words with respect to the ground-truth label, compared with baseline methods. As shown in Table 9, DroPTC-WoCW achieves the highest precision@k for $k = 1$, $k = 2$, and $k = 3$, demonstrating that fine-tuning embedding improves the semantic alignments, compared to the baselines. Fig. 10 shows a sample of problem log, to answer the question *which words in the input log are considered relevant to the label?* From the figure, it can be seen that DroPTC-WoCW more accurately pinpoints the relevant words, "stopped working", to the actual class, Software Issue. It can help the analyst closely monitor whether the model accurately captures log semantics.

5.2. Case study

Overview. This section answers the last research question. A case study was conducted on seven flights with various problem types covered in a real-world setting. In each flight log, we identified the unique problem-indicating logs, measured precision and recall against the key problem-indicating events identified by the expert shown in Table 3. Furthermore, effort reduction was measured by taking the ratio between the number of irrelevant events over all events within each flight log (Lin et al., 2016).

Problem identification performance. Table 10 summarizes the output of DroPTC in identifying problem-expressing logs within the seven case study flight logs. Overall, DroPTC successfully finds all problem-related log sentences, with a perfect recall and precision on the unique problematic log sentences, based on the expert analysis listed in Table 3. Interestingly, DroPTC manages to find problem-indicating log sentences that were not identified by the expert, which include `Downlink Lost`, `Aircraft and remote controller disconnected.`, and `No image transmission`. The complete list of found problems is presented in Table 11. Consequently, DroPTC demonstrates its practicality to assist investigations as a quick data triage and root cause analysis tool. Nevertheless, in terms of problem type accuracy, DroPTC

Table 10
The number of messages (#M), sentences (#S), problem-indicating sentences (#P), and unique problem-indicating sentences (#UP) found within each flights. The ↓ shows the workload reduction percentage. The precision and recall are computed only on the unique problem-indicating log sentences.

Flight	#M	#S	#P	#UP	Pre	Rec
1	22	45	15 (↓0.67)	3 (↓0.93)	1.00	1.00
2	133	392	152 (↓0.61)	6 (↓0.98)	1.00	1.00
3	186	593	188 (↓0.68)	5 (↓0.99)	1.00	1.00
4	34	126	48 (↓0.62)	6 (↓0.95)	1.00	1.00
5	75	186	68 (↓0.63)	6 (↓0.97)	1.00	1.00
6	58	276	96 (↓0.65)	6 (↓0.98)	1.00	1.00
7	150	540	185 (↓0.66)	10 (↓0.98)	1.00	1.00
Total	658	2158	752 (↓0.65)	42 (↓0.98)	–	–

Fig. 10. The word importance given by integrated gradient towards the Software Issue class. < s> and </s> are special tokens.

Table 11
The complete list of problem-indicating log sentences found in each flight. Gray highlight is given to incorrect predictions. Red highlights are the missed events by the expert manual analysis.

No.	Log Sentence	Problem Type	
		Predicted	Expert
1	Gimbal pitch axis endpoint reached.	Hw	Pv
	Image transmission signal weak.	Cm	Cm
	RC signal lost.	Cm	Cm
2	Downlink Lost.	Cm	Cm
	Gimbal pitch axis endpoint reached.	Hw	Pv
	Image transmission signal weak.	Cm	Cm
	No image transmission.	Cm	Cm
	RC signal lost.	Cm	Cm
	RC signal weak.	Cm	Cm
3	Downlink Lost.	Cm	Cm
	Image transmission signal weak.	Cm	Cm
	No image transmission.	Cm	Cm
	RC signal lost.	Cm	Cm
	RC signal weak.	Cm	Cm
4	Downlink Lost.	Cm	Cm
	Gimbal auto check failed.	Sw	Sw
	Gimbal stuck.	Hw	Hw
	Image transmission signal weak.	Cm	Cm
	RC signal lost.	Cm	Cm
	RC signal weak.	Cm	Cm
5	Ambient light too low.	Sr	Sr
	Compass error.	Hw	Hw
	Downward ambient light too low.	Sr	Sr
	Gimbal pitch axis endpoint reached.	Hw	Pv
	RC signal lost.	Cm	Cm
	RC signal weak.	Cm	Cm
6	Ambient light too low.	Sr	Cm
	Compass error.	Hw	Hw
	Downward ambient light too low.	Sr	Sr
	Forward ambient light too low.	Sr	Sr
	Low battery RTH.	Hw	Hw
	Low battery.	Hw	Hw
7	Aircraft and remote controller disconnected.	Hw	Cm
	Ambient light too low.	Sr	Sr
	Downlink Lost.	Cm	Cm
	Downward ambient light too low.	Sr	Sr
	Forward ambient light too low.	Sr	Sr
	Image transmission signal lost.	Cm	Cm
	Image transmission signal weak.	Cm	Cm
	No image transmission.	Cm	Cm
	RC signal lost.	Cm	Cm
	RC signal weak.	Cm	Cm

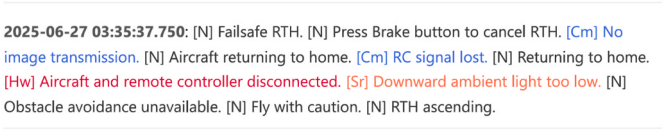

Fig. 11. A snapshot of the constructed timeline with highlights given to problem-indicating logs along with the predicted problem types.

incorrectly predicts `Gimbal pitch axis endpoint reached` and `Aircraft and remote controller disconnected` as Hardware Issue, while the expert labels them as Parameter Violation and Communication Issue, respectively.

Effort reduction. Reducing the analyst workload is one of the main objectives of forensic tools development. An analysis is conducted to quantify the amount of workload reduced with the help of DroPTC, as summarized in Table 10. After performing the problem identification, DroPTC found 752 problem-indicating logs out of 2158 total log events, reducing the analyst workload by up to 65 %. Due to the nature of logs that contain repetitive sentence patterns, we perform deduplication to remove duplicate events, leaving unique problem sentences, which are 42. At this stage, DroPTC successfully compresses the log volume from 2158 to only 42, reducing the analyst workload by up to 98 %. This demonstrates the usefulness of DroPTC as a data triage tool to eliminate irrelevant artifacts for the analyst to focus only on key events. To further assist an analyst, a forensic timeline is constructed in which problem-indicating events are highlighted, as depicted in Fig. 11. In addition,

Table 12
Number of unique template (#Unique), seen template during training (#Seen), and novel template (#Unseen) within each flight logs.

Flight	#Unique	#Seen	#Unseen
1	15	1 (6.7 %)	14
2	30	4 (13.3 %)	26
3	29	4 (13.8 %)	25
4	26	4 (15.4 %)	22
5	26	5 (19.2 %)	21
6	26	6 (23.1 %)	20
7	33	7 (21.2 %)	26

DroPTC produces various visualizations, such as an interactive timeline grouped by problem type, the top 20 most frequent log events, a timeline with highlights given to problem-indicating sentences, the problem type summary, and a heatmap of word importance.

Generalization analysis. DroPTC employs pre-trained embedding to exploit the semantics inherent in the corpora used during the pre-training stage. It makes it possible to train a problem classifier by using a small number of training samples. Given that DJI FPV is also covered in the training samples originating from AirData, we investigate and verify if the performance of DroPTC is due to memorization or generalizes well to unseen log structures. Table 12 presents the number of unique log sentence templates obtained from the Drain (He et al., 2017), seen templates during the training, and unseen log structures. It can be seen that most of the logs in the case study were not seen by DroPTC during training. Therefore, the performance is not merely due to memorization.

5.3. Discussion

DroPTC demonstrates promising performance in effectiveness, efficiency, and trustworthiness. Based on the experimental results, class weight can improve the accuracy. The calibration and word-importance analyses indicate that DroPTC-WoCW exhibits better calibration and more aligned semantics. To assess the trade-off between performance gains and trustworthiness, we conduct *t*-test tests across key metrics. The improvement in the F1 score attributable to the class weight, as shown in Tables 7, is not statistically significant ($p = 0.1147$). Conversely, DroPTC-WoCW improves the mean confidence score by 0.3728 ($p = 0.0011$), thereby improving the calibration, as shown in Fig. 8. A cross-run stability test further supports the dominance of DroPTC-WoCW, as depicted in Fig. 9. Additionally, the word importance shows higher precision on average, with a mean difference of 0.0801 ($p < 0.001$), as shown in Table 9. In conclusion, DroPTC-WoCW is suited for forensic applications due to its trustworthiness and competitive accuracy.

5.4. Limitations and threats to validity

Limited case study. The case study reported in this paper was conducted using a single drone model, the DJI FPV. Although the collected dataset covers five of the six problem types, the variations in log structures are small. DroPTC has a limitation: the logs must be human-readable, so that features extracted from the embedding models are better aligned.

Rule-based segmenter. The first threat to validity arises from the fact that the rules used to segment log messages into sentences are static, making DroPTC prone to log structure evolution. Since the segmenter is positioned in the pre-processing step, the accuracy and utility of the segmented logs depend heavily on the quality of the rules. To this end, we plan to employ a semantic-based segmenter in the future, allowing for more reliable and robust sentence-level analysis that is less susceptible to changes in log structure.

Data leakage. The embedding models used in this paper were pre-trained on large, general-purpose corpora. It is not guaranteed that, among the text in the corpora, there are no drone flight log messages. In

Table 13
Seen sentence patterns in the case study dataset. Highlights are given to the problem-indicating log sentence.

Sentence	Template
Aircraft braking.	Aircraft <*>
Data Recorder File Index is 83.	Data Recorder File Index is <*>
Downlink Lost.	Downlink Lost.
Downlink Restored (after 0m 6.2s).	Downlink Restored (after 0m <*>
Data Recorder File Index is 84.	Data Recorder File Index is <*>
Downlink Restored (after 0m 5.8s).	Downlink Restored (after 0m <*>
Downlink Restored (after 0m 7.4s).	Downlink Restored (after 0m <*>
Downlink Restored (after 0m 5.4s).	Downlink Restored (after 0m <*>
Data Recorder File Index is 88.	Data Recorder File Index is <*>
Downlink Restored (after 0m 5.2s).	Downlink Restored (after 0m <*>
Downward ambient light too low.	Downward ambient light too low.
Obstacle avoidance unavailable.	Obstacle avoidance unavailable.
Ambient light too low.	Ambient light too low.
Low battery.	Low battery.
Aircraft returning.	Aircraft <*>
Data Recorder File Index is 89.	Data Recorder File Index is <*>
Downlink Restored (after 0m 22.8s).	Downlink Restored (after 0m <*>
Downlink Restored (after 0m 16s).	Downlink Restored (after 0m <*>

other words, it is possible that the model has seen some of the log messages and structures. It can challenge the fairness of the evaluation. To this end, we perform generalization analysis by checking the overlapping samples between the training set and the case study set. Table 13 shows the list of seen log structures, where four of them are problem-indicating and appear in the case study.

6. Conclusion and future work

This paper proposes DroPTC, an end-to-end framework for identifying problem-indicating events in drone flight logs for forensic investigations. Instead of analyzing log records at the message level, which often contain multiple sentences, we propose a novel analysis perspective by segmenting log messages into individual sentences. This allows an analyst to pinpoint the log segment relevant to the investigation more accurately than message-level analysis. Contrastive fine-tuning is employed to align semantics with domain-specific context, thereby improving the model's performance and confidence and yielding admissible evidence. A sentence deduplication is used to identify unique events, thereby reducing the analyst's workload in investigating the relevant artifacts. Extensive experiments, a case study, and analyses demonstrate that DroPTC outperforms the baselines in effectiveness, efficiency, and trustworthiness. DroPTC has been implemented and evaluated as an open-source tool to support research in drone forensics.

Despite the promising results, DroPTC is prone to log evolution due to the static pre-processing rules. A semantic-based segmenter is one of the future directions we plan to pursue. Furthermore, we plan to include additional drone models in the case study to provide more representative evaluations.

Acknowledgment

This research is funded by the Indonesian Endowment Fund for Education (LPDP) on behalf of the Indonesian Ministry of Higher Education, Science and Technology and managed under the EQUITY Program (Contract No 4299/B3/DT.03.08/2025, No 3029/PKS/ITS/2025 & No 2345/IT2.III.1/T/KP.03.00/XII/2025).

Data availability

The source code, dataset, and figures are publicly available on GitHub: https://github.com/swardiantara/droptc-dev.

References

AirData UAV, I., 2023. Drone error and warning codes - world's Most comprehensive list. https://app.airdata.com/wiki/Notifications/.

Alam, S., Altiparmak, Z., 2024. XAI-CF – examining the role of explainable artificial intelligence in cyber forensics. https://arxiv.org/abs/2402.02452.

Alcántara, A., Capitán, J., Cunha, R., Ollero, A., 2021. Optimal trajectory planning for cinematography with multiple unmanned aerial vehicles. Robot. Autonom. Syst. 140, 103778. https://doi.org/10.1016/j.robot.2021.103778.

Ayamga, M., Akaba, S., Nyaaba, A.A., 2021. Multifaceted applicability of drones: a review. Technol. Forecast. Soc. Change 167, 120677. https://doi.org/10.1016/j.techfore.2021.120677.

Breton, L.L., Fournier, Q., Morris, J.X., Mezouar, M.E., Chandar, S., 2025. NeoBERT: a next generation BERT. Transact. Mach. Learn. 5, 1–23. https://openreview.net/forum?id=TJRyDi7mwH.

Capuano, N., Fenza, G., Loia, V., Stanzione, C., 2022. Explainable artificial intelligence in CyberSecurity: a survey. IEEE Access 10, 93575–93600. https://doi.org/10.1109/ACCESS.2022.3204171.

Cicek, D., Kantarci, B., Schillo, S., 2025. A comparative review of user acceptance factors for drones and sidewalk robots in autonomous last mile delivery. Green Energy Intell. Transp. 4, 100310. https://doi.org/10.1016/j.geits.2025.100310.

Clark, D.R., Meffert, C., Baggili, I., Breitinger, F., 2017. DROP (DRone open source parser) your drone: forensic analysis of the DJI phantom III. Digit. Invest. 22, S3–S14. https://doi.org/10.1016/j.diin.2017.06.013.

Devlin, J., Chang, M.W., Lee, K., Toutanova, K., 2019. BERT: pre-training of deep bidirectional transformers for Language understanding. In: NAACL-HLT, pp. 4171–4186. https://doi.org/10.18653/v1/N19-1423.

Editya, A.S., Ahmad, T., Studiawan, H., 2023. Forensic investigation of drone malfunctions with transformer. In: International Conference on Smart Systems for Applications in Electrical Sciences (ICSSES), pp. 1–5. https://doi.org/10.1109/ICSSES58299.2023.10199237.

Hadsell, R., Chopra, S., LeCun, Y., 2006. Dimensionality reduction by learning an invariant mapping. CVPR 1735–1742. https://doi.org/10.1109/CVPR.2006.100.

Hall, S.W., Sakzad, A., Choo, K.K.R., 2022. Explainable artificial intelligence for digital forensics. WIREs Foren. Sci. 4, e1434. https://doi.org/10.1002/wfs2.1434.

Hargreaves, C., Breitinger, F., Dowthwaite, L., Webb, H., Scanlon, M., 2024. DFPulse: the 2024 digital forensic practitioner survey. Forensic Sci. Int.: Digit. Invest. 51, 301844. https://doi.org/10.1016/j.fsidi.2024.301844.

He, P., Zhu, J., Zheng, Z., Lyu, M.R., 2017. Drain: an online log parsing approach with fixed depth tree. In: IEEE International Conference on Web Services (ICWS), pp. 33–40. https://doi.org/10.1109/ICWS.2017.13.

Josephat, A., Sekar, A., T, D., Angalaeswari, S., 2025. Design and development of agricultural drone for precision fertilizer application to optimize crop yields. Results Eng. 27, 106267. https://doi.org/10.1016/j.rineng.2025.106267.

Laricchia, F., 2022. Consumer drone unit shipments worldwide from 2020 to 2030. https://www.statista.com/statistics/1234658/worldwide-consumer-drone-unit-shipments.

Le, V., Zhang, H., 2021. Log-based anomaly detection without log parsing. In: 36th IEEE/ACM Intl. Conf. on Automated Softw. Eng. (ASE), pp. 492–504. https://doi.org/10.1109/ASE51524.2021.9678773.

Lin, Q., Zhang, H., Lou, J.G., Zhang, Y., Chen, X., 2016. Log clustering based problem identification for online service systems. In: Proceedings of the 38th International Conference on Software Engineering Companion, pp. 102–111. https://doi.org/10.1145/2889160.2889232.

Mantas, E., Patsakis, C., 2019. GRYPHON: drone forensics in dataflash and telemetry logs. In: Advances in Information and Computer Security, pp. 377–390. https://doi.org/10.1007/978-3-030-26834-3_22.

Mantas, E., Patsakis, C., 2022. Who watches the new watchmen? The challenges for drone digital forensics investigations. Array 14, 100135. https://doi.org/10.1016/j.array.2022.100135.

Minn, W., Tun, Y.N., Shar, L.K., Jiang, L., 2024. Dronlomaly: runtime log-based anomaly detector for dji drones. In: Proceedings of the 2024 IEEE/ACM 46th International Conference on Software Engineering: Companion Proceedings, pp. 6–10. https://doi.org/10.1145/3639478.3640042.

Mombelli, S., Lyle, J.R., Breitinger, F., 2024. FAIRness in digital forensics datasets' metadata – and how to improve it. Forensic Sci. Int.: Digit. Invest. 48, 301681. https://doi.org/10.1016/j.fsidi.2023.301681.

Pham, T.A., Lee, J.H., 2023. TransSentLog: interpretable anomaly detection using transformer and sentiment analysis on individual log event. IEEE Access 11, 96272–96282. https://doi.org/10.1109/ACCESS.2023.3311146.

Puchalski, R., Giernacki, W., 2022. UAV Fault detection methods, state-of-the-art. Drones 6. https://doi.org/10.3390/drones6110330.

Quero, C.O., Martinez-Carranza, J., 2025. Unmanned aerial systems in search and rescue: a global perspective on current challenges and future applications. Int. J. Disaster Risk Reduct. 118, 105199. https://doi.org/10.1016/j.ijdrr.2025.105199.

Reimers, N., Gurevych, I., 2019. Sentence-BERT: sentence embeddings using siamese BERT-networks. In: EMNLP-IJCNLP, pp. 3982–3992. https://doi.org/10.18653/v1/D19-1410.

Rjoub, G., Bentahar, J., Abdel Wahab, O., Mizouni, R., Song, A., Cohen, R., Otrok, H., Mourad, A., 2023. A survey on explainable artificial intelligence for cybersecurity. IEEE Transact. Netw. Serv. Manag. 20, 5115–5140. https://doi.org/10.1109/TNSM.2023.3282740.

Shar, L.K., Minn, W., Ta, N.B.D., Fan, J., Jiang, L., Kiat, D.L.W., 2022. DronLomaly: runtime detection of anomalous drone behaviors via log analysis and deep learning. In: 29th Asia-Pacific Software Engineering Conference (APSEC), pp. 119–128. https://doi.org/10.1109/APSEC57359.2022.00024.

Silalahi, S., Ahmad, T., Studiawan, H., 2023. DroNER: dataset for drone named entity recognition. Data Brief 109179doi. https://doi.org/10.1016/j.dib.2023.109179.

Silalahi, S., Ahmad, T., Studiawan, H., Anthi, E., Williams, L., 2024. Severity-oriented multiclass drone flight logs anomaly detection. IEEE Access 12, 64252–64266. https://doi.org/10.1109/ACCESS.2024.3396926.

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 13, 105361–105379. https://doi.org/10.1109/ACCESS.2025.3580056.

Studiawan, H., Grispos, G., Choo, K.K.R., 2023. Unmanned aerial vehicle (UAV) forensics: the good, the bad, and the unaddressed. Comput. Secur. 132, 103340. https://doi.org/10.1016/j.cose.2023.103340.

Sun, J., Hubbard, S., 2025. An examination of UAS incidents: characteristics and safety considerations. Drones 9. https://doi.org/10.3390/drones9020112.

Sun, Z., Ruan, N., Li, J., 2025. DDL: effective and comprehensible interpretation framework for diverse deepfake detectors. IEEE Trans. Inf. Forensics Secur. 20, 3601–3615. https://doi.org/10.1109/TIFS.2025.3553803.

Sundararajan, M., Taly, A., Yan, Q., 2017. Axiomatic attribution for deep networks. In: Proceedings of the 34th International Conference on Machine Learning, 70, pp. 3319–3328.

Tan, I., Minn, W., Poskitt, C.M., Shar, L.K., Jiang, L., 2025. Runtime anomaly detection for drones: an integrated rule-mining and unsupervised-learning approach. In: Engineering of Complex Computer Systems: 29Th International Conference, ICECCS 2025. Proceedings, pp. 3–23. https://doi.org/10.1007/978-3-032-00828-2_1.

Wang, D., Li, S., Xiao, G., Liu, Y., Sui, Y., He, P., Lyu, M.R., 2024. An exploratory investigation of log anomalies in unmanned aerial vehicles. In: Proceedings of the IEEE/ACM 46th International Conference on Software Engineering, pp. 1–13. https://doi.org/10.1145/3597503.3639186.

Warner, B., Chaffin, A., Clavié, B., Weller, O., Hallström, O., Taghadouini, S., Gallagher, A., Biswas, R., Ladhak, F., Aarsen, T., Adams, G.T., Howard, J., Poli, I., 2025. Smarter, Better, Faster, Longer: a Modern Bidirectional Encoder for Fast, Memory Efficient, and Long Context Finetuning and Inference. ACL, pp. 2526–2547. https://doi.org/10.18653/v1/2025.acl-long.127.

Zhang, Z., Hamadi, H.A., Damiani, E., Yeun, C.Y., Taher, F., 2022. Explainable artificial intelligence applications in cyber security: state-of-the-art in research. IEEE Access 10, 93104–93139. https://doi.org/10.1109/ACCESS.2022.3204051.