Hybrid offline-online machine learning framework for real-time UAV battery voltage prediction

Aerospace Science and Technology 177 (2026) 112217

Contents lists available at ScienceDirect

Aerospace Science and Technology

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

Original article

Hybrid offline-online machine learning framework for real-time UAV battery voltage prediction

Leonardo Baldo[†] [a,*], Jorge E. García Bustos[†] [b], Benjamin Brito Schiele [c], Ricardo Salas-Espiñeira [b], Andrea De Martin [a], Marcos E. Orchard [b]

[a] Department of Mechanical and Aerospace Engineering, Politecnico di Torino, Corso Duca degli Abruzzi 24, Torino, 10129, Italy
[b] Department of Electrical Engineering, Faculty of Physical and Mathematical Sciences, University of Chile, Av. Tupper 2007, Santiago, Chile
[c] Intelligent System Prognostics Group, Aerospace Structures and Materials Department, Faculty of Aerospace Engineering, Delft University of Technology, the Netherlands

ARTICLE INFO

Chaoyong Li

Keywords:
Unmanned aerial vehicles
Drones
Energy prediction
Physics-based battery modeling
GRU
CRPS

ABSTRACT

Unmanned Aerial Vehicles (UAVs) are increasingly adopted in logistics and mission-critical applications, where reliable battery state-of-energy information is essential for safe planning and execution. However, robust voltage prediction for real-time operations remains underexplored, and existing approaches have not explored the possibility of combining offline prediction with discrepancy-triggered in-flight updates to handle the uncertainty raising from the lack of knowledge about future power profiles. In this sense, this paper proposes a dual-phase voltage prediction framework that (i) provides a first estimate of the voltage trend along a planned route using an offline multi-criteria segmentation of historical flights and a route-matching procedure, and (ii) refines predictions in-flight through a discrepancy-triggered update based on the Continuous Ranked Probability Score (CRPS), rerunning the model only when significant deviations are detected to limit computational burden and to adapt to unseen mission profiles. The method combines a hybrid physics-based/data-driven formulation with a Gated Recurrent Unit (GRU) trained on up to 200 flights to model battery behavior and to forecast the full voltage trajectory. Validation on a publicly available dataset shows that the proposed offline-online strategy substantially improves prediction accuracy, reducing MAE from 0.181 V to 0.087 V (51 % improvement). A sensitivity analysis on the CRPS threshold parameters highlights the trade-off between error reduction and computational cost. The overall work incorporates innovative methodologies with a focus on practical applicability, thus facilitating the advancement of ML strategies in these domains.

1. Introduction

From maintenance inspections to aerial photography, from urban mobility to package delivery, from Search and Rescue (SAR) missions to surveillance tasks, the deployment of Unmanned Aerial Vehicles (UAVs) in operational scenarios is leaving a permanent mark in a wide range of civil and military applications [1,2]. However, the disruptive advantages provided by the deployment of UAVs present a consistent set of challenges that must be addressed to enable the seamless integration of UAVs into everyday applications. Among these challenges, the development of data-to-strategy methodologies for online battery state monitoring is regarded as one of the most demanding and immediate issues to address, as it directly impacts the platform reliability and safety [3]. For example, during infrastructure inspections in remote or inaccessible areas, UAV deployments are often employed. This is the case of the structural monitoring of the Atacama Large Millimeter/submillimeter Array (ALMA) facilities, located in San Pedro de Atacama, Chile. In these challenging environments, UAVs offer an optimal balance between operational availability and inspection quality [4]. Monitoring battery performance and endurance is particularly important in such contexts, as factors such as altitude and significant temperature fluctuations can substantially affect battery energy consumption [5].

Battery state monitoring typically spans two temporal scales: the State of Health (SOH), which assesses the overall health status of the battery, and the State of Energy (SOE), which evaluates the current charge level of the battery. If effective battery health monitoring is essential to foster UAV acceptance as reliable tools for urban mobility and critical missions, accurate assessment and management of battery energy state based on online data-driven predictive methods are crucial to ensure safe flight operations and successful mission completion. On the

† These authors contributed equally to this work and share first authorship.

* Corresponding author.
E-mail addresses: leonardo.baldo@polito.it (L. Baldo), jorgegarcia@ug.uchile.cl (J.E. García Bustos), benjamin.brito@ug.uchile.cl (B. Brito Schiele), ricardo.salas.e@ug.uchile.cl (R. Salas-Espiñeira), andrea.demartin@polito.it (A. De Martin), morchard@ing.uchile.cl (M.E. Orchard).

https://doi.org/10.1016/j.ast.2026.112217
Received 30 November 2025; Received in revised form 3 March 2026; Accepted 23 March 2026
Available online 29 March 2026

one hand, Battery SOH monitoring is a highly developed and specialized area of research, with papers focusing on the impact of temperature and charge-discharge cycles on the overall SOH of cells [6–8]. These research efforts translate into many studies that focus specifically on UAV operations, where the SOH is studied through deep Machine Learning (ML) pipelines and physics-based models [9].

On the other hand, since UAV operability is significantly affected by energy-intensive operations, accurate prediction of SOE consumption is essential to prolong flight durations [10], optimize mission planning, and support sustainable operational practices [11–13]. The two lines of research on SOH and SOE must go hand in hand to advance the current state-of-the-art. However, progress in the development of SOE models is still impeded by the absence of standardized methods and frameworks to estimate real-time energy consumption and assess the remaining range and endurance. Even under the assumption that no battery faults occur, accurately monitoring and predicting SOE (and the associated voltage trend) is essential for mission safety. In fact, safety-critical constraints in electric UAV operations are often driven by energy margins and voltage limits rather than by fault scenarios. It is evident that an in-flight emergency resulting from insufficient power due to battery discharge could ultimately result in the loss of the UAV, with potentially catastrophic consequences. For this reason, SOE/voltage forecasting supports (i) mission planning with adequate reserves (e.g., return-to-home, alternate landing, contingency maneuvers), (ii) anticipation of low-voltage operating regions, and (iii) timely operational decisions such as route shortening or early return before entering a critical energy state. These factors, combined with the currently limited absolute range provided by state-of-the-art batteries, underscore the relevance of addressing the existing research gaps in energy prediction models for battery-related UAVs. In fact, despite their importance, the current body of research on energy consumption models remains limited, especially given the considerable advantages that UAV operations could provide in the short to medium term [14]. As a consequence, in this paper we focus on SOE/voltage prediction at the mission time scale; battery aging (SOH) evolves on a much slower time scale across many flights/cycles and is therefore outside the scope of the present study. Given the rapid evolution of the field, we decided to adopt standard terminology: UAV denotes the aircraft itself, whereas Unmanned Aircraft System (UAS) refers to the UAV together with the associated elements required for operation. The term drone is used as a widely adopted, non-technical synonym in the broader literature and, in the context of this work, refers to the same UAV platforms considered in our experiments.

1.1. Research objectives and contributions

In view of the extensive potential of UAV operations, the authors propose a modular hybrid real-time UAV battery voltage prediction framework, which combines an offline estimation phase with an online update phase, capable of; supporting the deployment of UAVs in operational scenarios, providing a way to increase the social acceptance of UAVs in urban applications by addressing the challenge of unknown future power profiles. Our key contributions are as follows:

- We propose an innovative, dual-phase framework specifically tailored to address the uncertainty inherent in UAV future power demand. To effectively manage this uncertainty, the framework (i) first generates an a priori voltage trend estimation using an offline methodology, and (ii) subsequently refines these predictions in real time during the flight. This online refinement is triggered only when a discrepancy metric (Continuous Ranked Probability Score, CRPS) indicates a significant deviation, resulting in a robust and computationally efficient adaptation to the actual mission profile.
- Our work incorporates a multi-criteria segmentation protocol that partitions historical flight data into dynamically uniform segments, based on lumped linear-acceleration peaks and time customizable thresholds. This dual trigger captures both active maneuvering and hovering periods throughout the mission profile.
- We implement an offline estimation based on Dynamic Time Warping (DTW) route-matching procedure to generate a-priori voltage prediction. During the flight, the CRPS metric triggers voltage trends updates only when meaningful deviations with respect to the prior trend are detected, resulting in a computationally efficient online prediction.
- We select and adopt an hybrid multi-physics battery model to accurately simulate battery discharge. Specifically, the framework employs a ML data-driven core fueled by two bidirectional Gated Recurrent Units (GRUs) and a physics based battery model representation using a Thévenin circuit. This approach allows the model to capture both battery behaviors and physical dynamics.

The remainder of the paper is organized as follows. Section 2 provides an overview of the current state-of-the-art models for the prediction of UAV energy and voltage, highlighting a substantial research gap. Section 3 offers an overview of the framework, including a brief description of the methodologies used and an introduction to the selected ML architecture. Section 4 details the description of the dataset, as well as the development and results of the main ML model. Following this, Section 5 presents the framework details, including the integration of the ML model into the overall strategy. Section 6 presents the results of the simulations conducted within the entire framework. Finally, Section 7 summarizes the conclusions and outlines potential directions for future research.

2. Background and problem statement

Fueled by the ever-growing interest towards the reduction of Green House Gases (GHG), the study of remaining range predictions for Electric Vehicles (EV) is a consolidated and high-paced growing research field with a plethora of possible implementations and strategies. The results of this enormous research effort towards de-carbonization in e-mobility are being reflected in the steep rise of electric powered vehicle acquisitions and EV integration in many fields and applications. Despite sharing the same technology base, the development of energy consumption models and endurance estimation methodologies for battery-powered UAVs represents a distinct narrative.

In fact, the research body on UAVs is scattered and is held back by the scarcity of publicly available and comprehensive datasets, replicability and comparison challenges, and the need to deal with ever-changing operational conditions [15]. Moreover, most of the retrieved records depict a scenario in which the main objective of the research paper is to develop a higher level comparison between the use of UAVs in lieu of conventional means of transportation in the context of GHG reduction [16], without a solid focus on modeling battery energy consumption.

Despite the limited number of papers, existing methodologies for predicting energy consumption in UAVs can be divided into a set of four classes. Taking inspiration from the categories of energy consumption models presented in the excellent work of Zhang et al. [17] and Cabuk et al. [18], the authors divided the models from the existing literature into four main classes. In fact, apart from simple modeling strategies that assume that energy consumption is constant per unit of time or traveled distance [19] (e.g., [20,21]), the literature approaches can be roughly divided into: Physics-Based, Regression, Datasheet and ML powered models. Physics-Based models leverage flight mechanics equations from fixed or rotary wing platforms to derive solid and mathematically sound strategies for endurance and range estimations [22,23]. Regression models exploit experimental data to obtain a set of polynomial equations which fit experimental data trying to extract real-life behavior and condensate it into rigid mathematical formulations along with validity ranges [24,25]. Datasheet models try to develop energy consumption models given the energy consumption indication provided by component manufacturers [26]. Finally, ML powered models use

state-of-the-art strategies and methods to create models trained on experimental data [18,27–29].

There is a consensus among the research community that five methodologies represent the grounding stones for UAV energy consumption models. These methodologies have been developed following the Physics-based or Regression methodology. Four of these studies approach the UAV energy consumption challenge taking inspiration from fixed-wing flight mechanics approaches or more complete approaches which consider the forces acting on the UAV. For example, D'Andrea [30] presented one of the first models to use the effect of a single parameter inspired by fixed-wing platforms. This approach has been then improved by Figliozzi [31]. On the other hand, Dorling et al. [5], Stolaroff [32], and Kirschstein [33] present methodologies based on the forces acting on the UAV. Finally, the Tseng Model (2017) is based on a set of regression equations which have been finely tuned thanks to experimental data.

These approaches, despite providing solid groundwork for future research, still present challenges as the application in real-life scenarios is limited and constrained due to the assumptions on the formulas used and limited application range of regression equations. Having said that, in the last years, a few studies managed to apply ML principles to develop UAVs energy consumption models. To the best of our knowledge, the number of relevant studies is limited, and the main contributions are summarized in the following paragraphs.

Muli et al. [27], after a thorough description of existing methodologies, presented a Long-Short-Term Memory (LSTM) model that uses experimental training data. In particular, the authors in [27] have used 21 features in a stacked bidirectional LSTM-based architecture (BiLSTM). Moreover, a sensitivity analysis showed the extreme high importance of payload and the lower relevance of altitude and speed as features. Cabuk et al. [18], along with a theoretical and multivariate linear analysis, proposed a model based on the XGBoost Regressor algorithm and provided additional information in the case of UAV in the case of UAV swarms swarms. In particular, data processing with the XGBoost Regressor is preceded by an AutoEncoder (AE) to reduce dimensionality and highlight the most relevant features. Moreover, a feature importance analysis highlighted the weight as the most influential factor, followed by the speed, in accordance with Muli et al. [27] and Thibbotuwawa et al. [34]. The study carried out by Sarkar et al. [28] provides useful benchmark results on ML strategies for UAV energy consumption models. In particular, the authors in Sarkar et al. developed several ML models trained on the dataset provided by Rodrigues et al. [35]. Among the developed ML algorithms (Random Forest (RF), Regression Tree (RT), Support Vector Machine (SVM), Artificial Neural Network (ANN), and Adaptive Neuro-Fuzzy Inference System (ANFIS)), RF provided the best results in terms of Mean Absolute Error (MAE) and Root Mean Square Error (RMSE) values. Finally, El-Latif and El-dosuky [29], along with the proposal of two physics based models, presented a ML approach based on a random forest regressor especially tuned by applying the Fick's law algorithm optimizer.

Furthermore, the latest state-of-the-art has rapidly expanded to include advanced deep learning architectures for real-time UAV applications, including Recurrent Neural Networks (RNNs) and GRUs that capture temporal dependencies in complex flight data and improve trajectory prediction scalability [4,36].

To safely evaluate these energy-aware algorithms and path planning strategies before real-world deployment, recent research has pioneered advanced semi-physical simulation frameworks. A study by Lin et al. [37] constructed an improved six-axis semi-physical simulation system integrated with a physical mountainous environmental sand table. By fusing a deterministic energy-consumption prediction model with an Improved Adaptive Particle Swarm Optimization (IAPSO) algorithm, this framework effectively evaluates path planning while bridging the sim-to-real gap, perfectly contextualizing the need for robust, hardware-in-the-loop energy validation.

These advances confirm that, despite the scarce number of papers, ML estimators already quantify mission-level energy with good accuracy. Therefore, a logical progression is to complement this capability by predicting the entire voltage profile, estimating the terminal voltage at every point along the flight. In line with this transition, recent UAV research has begun to predict the complete terminal voltage trajectory rather than limiting the analysis to mission-level energy. A lightweight equivalent-circuit model augmented with a temporal temperature factor keeps voltage errors below 1.5 % [38]. An unscented Kalman filter configured for a 2 RCPNGV network, combined with an adaptive gain rule, maintains voltage errors derived from the State of Charge (SOC) within 1.0 % during aggressive maneuvers [39]. Despite these advances, both studies depend on single-cell laboratory data, omit the long-term effects of cycling degradation, and have yet to demonstrate robustness under rapid load swings typical of field operations.

In the broader literature on e-mobility, a neural mapping of the static SOC and V_{oc} curve coupled with an online teaching-learning optimizer cuts RMSE to less than 30 mV in automotive drive cycles [40]. Explicit temperature dependence embedded in Thévenin open-circuit voltage reduces the identification error to about 1 % without increasing the model order [41]. Deep learning approaches go further: an LSTM remains accurate after synthetic aging, with median errors less than 13 mV [42]; a temporal convolution network with attention, optimized via an improved whale-optimization algorithm, brings RMSE below 8 mV [43]; and a BiLSTM with multihead attention fused to a second order RC model reaches $R^2 = 0.91$ and MAPE under 1 % for joint voltage and SOC prediction [44]. Finally, an LSTM battery emulator trained on only five charge-discharge records reproduces large automotive cells with $R^2 = 0.97$ [45]. Yet these EV-oriented studies assume ground-vehicle load profiles, mild ambient variation, and well-regulated thermal management; none addresses the weight-energy trade-offs or the fast current spikes that characterize small multirotor platforms, nor the highly dynamic load profiles characterized by rapid, high-amplitude current spikes required for vertical liftoff and hovering.

To fill this gap, this work proposes a voltage prediction framework explicitly designed around the UAV operational scenario and to address the characterization of future power demand. Compared with ground EVs, UAV missions combine 3D kinematics, wind and environmental disturbances, and control-driven operating modes that generate highly non-stationary, rapidly varying load profiles. In this setting, prediction errors can directly compromise thrust margin and flight safety. To address this, we propose an offline-online strategy that (i) learns UAV-specific dynamic load patterns from historical flight data to produce an initial voltage forecast and (ii) refines that forecast in flight to account for real-time stochastic deviations in the future power profile. This two-step design is an adaptation developed to manage the uncertainty in UAV voltage prediction, thereby differentiating our contribution from prior works.

3. Framework overview & concept of operation

The framework implements a dual-phase voltage prediction strategy that includes an offline estimation phase using historical flight data combined with a ML core and an online update phase that refines the predictions by running the ML model leveraging updated real-time flight measurements. Beforehand, the ML models have been trained in an algorithm development phase, where historical flights are segmented using a multi-criteria protocol to create an organized training dataset. These flight segments train the ML core, which consists of two parallel GRUs that predict energy and current respectively, as reported in Fig. 1.

On the other hand, the dual phase process is illustrated in Fig. 2. The offline phase effectively integrates ML strategies with physics-based models in an actionable, customizable, and modular methodology [46]. In the offline phase, when a new flight is planned (i.e., trajectory in the 3D space is known), a route-matching procedure identifies similar

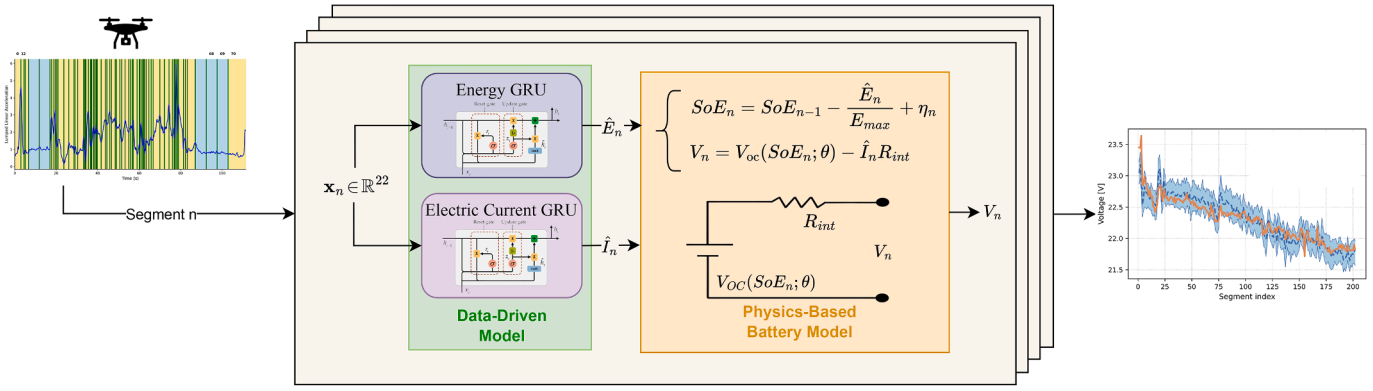


Fig. 1. Algorithm-development workflow used to train the proposed battery-voltage predictor.

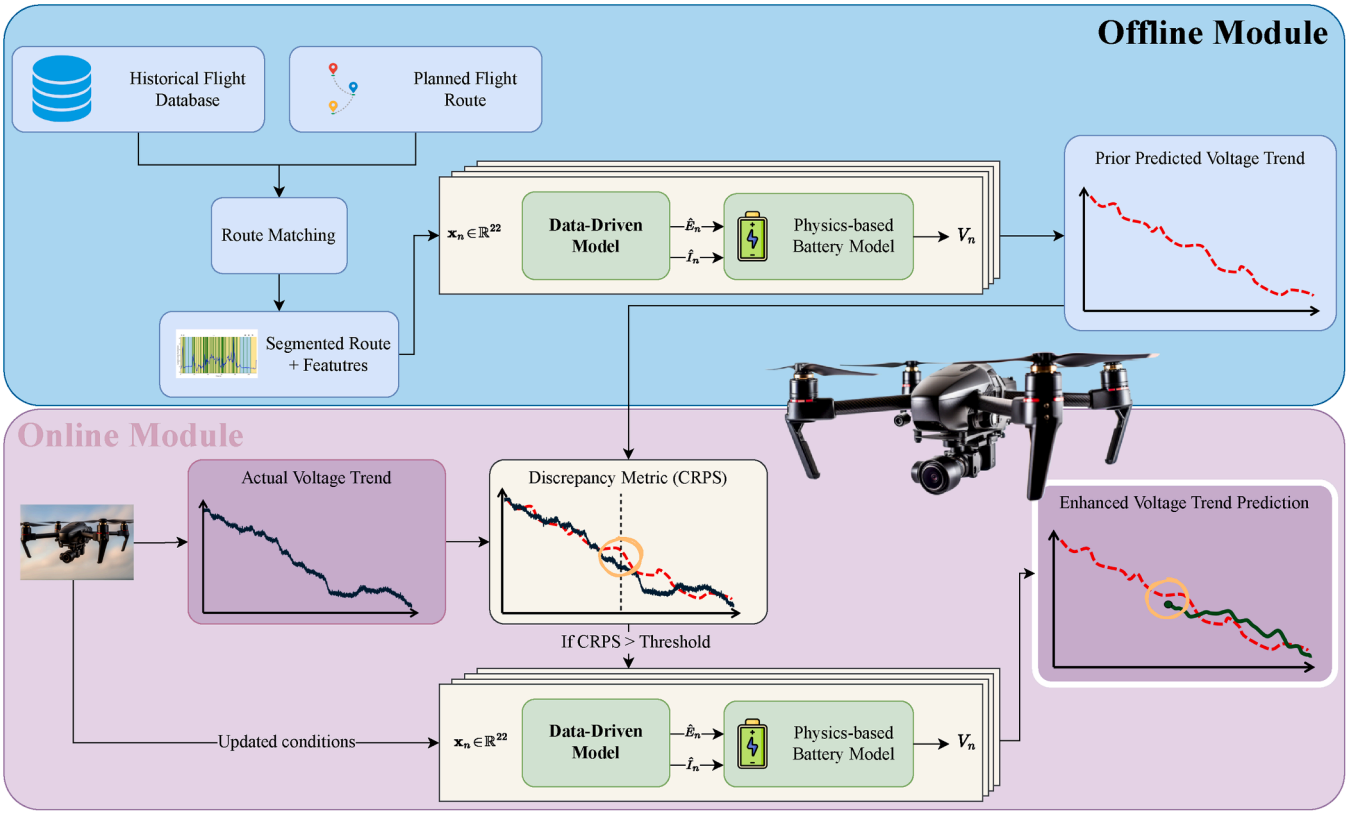


Fig. 2. Dual-phase voltage prediction framework combining an offline planning stage with an online adaptive update.

flight path by comparing the planned route against the historical routes database, to obtain a first estimation segmented flight. The pre-trained model then processes the identified segments to ultimately generate an initial voltage trend estimation through physics-based battery modeling. During flight execution, in the online phase, the initial estimation is continuously compared against actual voltage measurements and corrected when necessary by rerunning the ML core with updated real measurements. Updates are triggered by a discrepancy-driven strategy that uses the CRPS metric to determine when corrections are needed. As a secondary function, the discrepancy-driven strategy assists in mitigating the impact of noisy voltage sensors, which can result in inaccurate point values. In fact, this approach enables quick identification and correction of any divergence. The overall framework has been envisioned to improve the current voltage prediction when new field data are available. In particular, this study applies a specific mission processing protocol to systematically divide mission data into segments based on a multi-criteria strategy, resulting in intervals characterized by relatively uniform dynamics characteristics. The segmented tracks can then be utilized to model arbitrary missions that have not yet been traveled, predicting vehicle behavior for each segment based on data from similar UAV missions. In addition, the update phase drastically improves the ability of the models to generalize the predictions for energy consumption and voltage, resulting in a suitable methodology where limited historical training data are available.

In the context of this study, we assume that a three-dimensional trajectory flight plan is available before flight. For recreational UAVs, flight plans are typically not prepared in advance. However, for more mission-critical operations, a flight plan, including details such as trajectory, weight, speeds, and other relevant data, is often reviewed prior to the flight [47]. For instance, military UAV operations require comprehensive pre-flight planning, including detailed mission planning with coordinated flight routes. Similarly, commercial UAV delivery services operating under FAA Part 135 certification must file detailed flight plans and obtain route approvals before performing operations beyond visual line

of sight (BVLOS) [48]. As a result, this pre-planning requirement, far from being a limitation, aligns with established operational protocols in critical applications [49]. The details of each phase will be provided in Section 5, while Section 4.2 discusses the development of the algorithms used in each phase.

4. Dataset exploration and methodology

4.1. UAV typical mission and datasets

A typical UAV mission profile for a multipurpose/multi-mission UAV presents a take-off phase, followed by a climb and cruise phases. Each mission may include one or more specific submissions, each comprising Hover/Loiter, ingress, action, and egress phases. The mission ends with the cruise, descent, and landing phases. The final landing location may or may not be the same as the take-off location. During the action phase, the designated mission tasks are performed (e.g., package delivery, photos, etc.).

An extensive literature review was conducted to identify datasets representative of UAV missions and suitable for energy/voltage modeling. As summarized in Table 1, seven candidate datasets were identified [18,35,50–54]. However, several options present key limitations for the scope of this work: the lab-only battery cycling dataset [50] provides high-quality cell measurements but no UAV flight dynamics; some flight datasets are small in scale and there are not enough flights to train ML models [51,52]; and others either lack battery time-series entirely [54] or provide only partial battery information, without full voltage/current trajectories [18,53]. For these reasons, the dataset by Rodrigues et al. [35] was selected, as it is the most complete and informative: it contains the largest number of real, representative flights (209), includes the richest combination of signals (both UAV dynamics and battery time-series such as voltage/current/temperature), and provides sufficient metadata to fully parameterize and model the battery, with detailed information available for both the drone platform and the battery system.

The selected data set involves 209 flights performed by a DJI Matrice 100 quadcopter, equipped with a 4500 mAh battery, resulting in a weight of 3.68 kg excluding payload. The mission data, recorded at 1 Hz, include detailed records from a variety of sensors that span a total of almost 11 hours and 65 km. Data from Global Positioning System (GPS), the Inertial Measurement Unit (IMU), voltage and current sensors, an ultrasonic anemometer, along with metrics related to power consumption and battery voltage, were continuously monitored during flights, resulting in a total of 28 recorded signals. The UAV performed a total of 195 flights under various operational conditions, characterized by altitudes ranging from 25 to 100 m, velocities ranging from 4 to 12 m per second, and payload weights of 0 g, 250 g, and 500 g. Furthermore, 14 additional recordings were made during hovering or idle states of the UAVs, resulting in a cumulative total of 209 complete sorties.

4.2. ML algorithm development

This section details the modular pipeline to develop the ML model later employed in the use of the framework.

4.2.1. Data preprocessing

4.2.1.1. Training Segmentation Strategy. The flight profile has to be divided into segments to characterize the mission and to provide an organized training data set. In fact, to accurately model future mission data, there is the need of fine-tuned intervals from which a future mission could be modeled. It is important to divide the mission into segments that are sufficiently small to be treated as lumped points while also ensuring that the total number of segments remains manageable to limit the overall computational cost. A first idea would be to segment the flight into static phases (e.g., take-off, cruise, landing). However, this approach, despite being able to catch specific phase peculiarities, would

Table 1
Comparison of candidate datasets for UAV energy/voltage modeling on common grounds. Datasets are compared by acquisition setting (real flights vs. lab tests), vehicle/platform, configuration (Fixed-Wing (FW) vs. Rotary-Wing (RW)), dataset scale, intended objective/mission profile, and availability of flight dynamics and battery measurements. "Batt." indicates the presence of battery-related time series (e.g., voltage/current/temperature), where the asterisk denotes incomplete reporting. The bold line indicates the chosen dataset.

Ref.	Type	Vehicle / platform	FW/RW	Scale	Objective	Dyn.	Batt.	Notes
[35]	**Flights**	**DJI Matrice 100**	**RW**	**209 flights**	**Package delivery**	✓	✓	**UAV dynamics and time-series voltage/current on representative operational profiles.**
[50]	Lab	18,650 Li-ion cells	–	21,392 cycles	eVTOL mission profiles	×	✓	High-quality cell-level voltage/current/temperature; lacks UAV flight dynamics.
[51]	Flights	DJI Matrice 300	RW	7 flights	Waypoint flights (polygons)	✓	✓	Very low numery of flights train energy/voltage models.
[52]	Flights	DJI Pilot 2	RW	20 flights	Waypoint flights (polygons)	✓	✓	Low numery of flights train energy/voltage models
[54]	Flights	ROCON (custom)	RW	12 flights	Custom missions	✓	×	UAV dynamics available; no battery-related time-series (voltage/current) and limited battery information.
[18]	Flights	ModalAI m500	RW	98 flights	Custom missions	✓	✓*	UAV dynamics available; battery-related data reported only partially (no full time-series voltage/current).
[53]	Flights	MakeFlyEasy Fighter	FW	75 flights	Package delivery	✓	✓*	UAV dynamics available; battery-related data reported only partially (no full time-series voltage/current).

lead to poor prediction accuracy and low across-platform transferability as the segments would be extremely large. This led the authors to develop a custom multi-criteria segmentation protocol explained in the following paragraphs and presented in the pseudo-code reported in Algorithm 1. In the context of this paper, a segment is defined as a subset of continuous flight data points (samples). The segmentation protocol aims to divide data into distinct segments where flight dynamics conditions and hence energy consumption remain somewhat stable. Each segment contains a variable number of samples according to the specific segment. For small and medium UAVs, the motors are the main source of energy consumption and their power usage is closely linked to flight dynamics. To identify flight segments with similar flight dynamics, the acceleration trend and the time since the last trigger in the segment are considered (Fig. 4). In particular, the linear acceleration from the three-body axis has been lumped into a single value according to Eq. 1.

$$a_{lump} = \sqrt{a_x^2 + a_y^2 + a_z^2}. \tag{1}$$

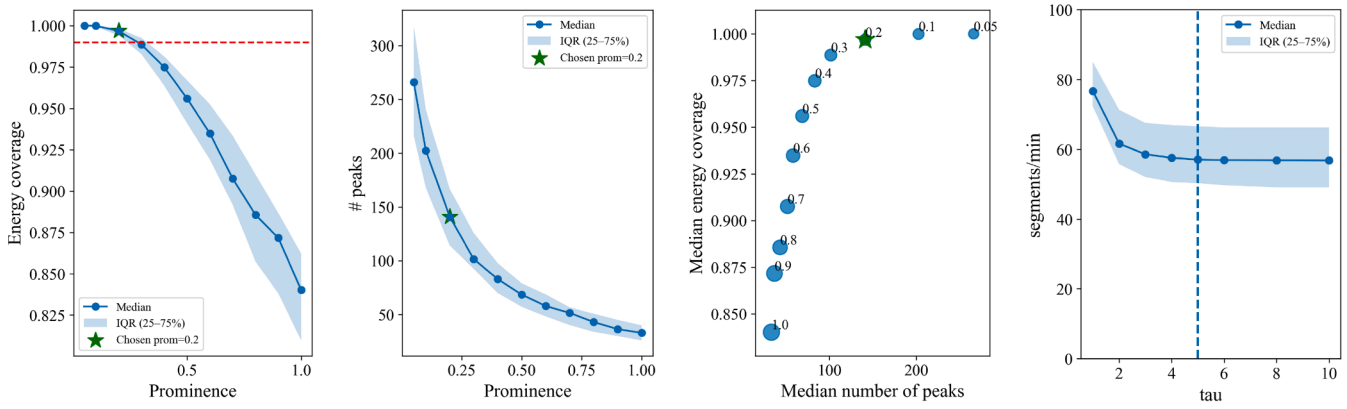
The lumped acceleration trend a_{lump} has then been filtered with a rolling window to remove noise [55], and then the peaks have been identified using the findpeaks function of the Python Scipy package, which finds all local maxima by comparing neighboring values with a set value of prominence (i.e., 0.2). Other more advanced peak detection strategies have been taken into consideration, but, given its simplicity, explainability, and low computational cost, this strategy has been deemed the most suitable. In this context, prominence quantifies the degree (salience) to which a peak stands out relative to its neighboring features. Specifically, it is defined as the vertical distance between the peak's height and the higher of the two adjacent local minima that bound it. Intuitively, a peak with high prominence clearly rises above the local baseline. Adjusting the prominence threshold allows for the filtering of minor oscillations or noise, thereby prioritizing the detection of more significant and salient events. A sensitivity analysis was conducted to determine the optimal prominence value with 50 randomly-selected flights from the dataset. The results are presented in Fig. 3 and support the selection of the prominence threshold by explicitly emphasizing fidelity and information preservation. We first quantify how much of the relevant dynamics is preserved after peak-based selection through an energy coverage metric, computed as the fraction of signal energy retained by the selected dynamics over the total energy of the considered dynamic component:

$$\mathrm{Cov}_E(p) = \frac{\sum_t d_{y,2}^{(p)}(t)^2}{\sum_t d_{y,2}(t)^2},$$

where $d_{y,2}(t)$ denotes the target dynamic signal (and $d_{y,2}^{(p)}(t)$ its reconstruction/selection induced by prominence p). Since the goal is to keep essentially all informative content, we impose $\mathrm{Cov}_E \geq 0.99$. Fig. 3a shows that this requirement is already satisfied around $p = .2$, with median coverage remaining very close to 1 (0.997). Given guaranteed fidelity, we subsequently select the simplest configuration among those that satisfy the coverage constraint. Fig. 3b show that increasing p substantially reduces the number of detected peaks, thereby suppressing weak fluctuations and mitigating over-segmentation; around $p = .2$ the reduction is pro-

Algorithm 1 Multi-criteria Segmentation Strategy.

Require: Subset of raw flight data $\{\mathbf{a}_t, \mathbf{x}_t, t\}_{t=1}^{T}$, where $\mathbf{a}_t = (a_x, a_y, a_z)$ and $\mathbf{x}_t$ includes all other flight variables
Require: Segmentation time threshold $\Delta T = 5$ s
Initialize segment list $\mathcal{S} \leftarrow \emptyset$
Initialize average data list $\mathcal{A} \leftarrow \emptyset$
Initialize last segment time $t_s \leftarrow 0$
Compute acceleration magnitude $a_t = \|\mathbf{a}_t\|_2$ for all t
Apply rolling mean filter to $\{a_t\}_{t=1}^{T}$ to obtain smoothed signal $\{\tilde{a}_t\}_{t=1}^{T}$
Detect peaks $\mathcal{P} = \mathrm{find\_peaks}(\tilde{a}_t)$ using SciPy
for $t \leftarrow 1$ **to** T **do**
  if $t \in \mathcal{P}$ **or** $t - t_{\text{last}} \geq \Delta T$ **then**
    Append t to $\mathcal{S}$
    Extract flight data in segment: $\mathcal{D}_{t_s:t+1}$
    Compute energy and average values $\bar{\mathbf{x}}_{t_s:t+1} = \mathrm{mean}(\mathcal{D}_{t_s:t+1})$
    Append $\bar{\mathbf{x}}_{t_s:t}$ to $\mathcal{A}$
    Update $t_s \leftarrow t$
return segmentation points $\mathcal{S}$ and segment-wise averages $\mathcal{A}$



(a) **Energy coverage.** Median energy coverage with IQR. Coverage remains near unity around $p = 0.2$, indicating preserved dominant dynamics. (b) **Number of peaks.** Median detected peaks with IQR. Increasing p removes weak/spurious peaks. (c) **Pareto trade-off.** Coverage vs complexity; marker size ∝ median segment duration. (d) **Time threshold.** Sensitivity of segmentation density to the time threshold.

Fig. 3. Prominence and τ sensitivity analysis across 50 flights. Prominence p controls the minimum peak salience retained in the findpeaks function. Increasing p reduces peak count, mitigating over-segmentation and spurious detections. Around $p = .2$, energy coverage remains near-complete, and the Pareto view confirms a knee-region trade-off between fidelity and complexity. The last subplot shows the sensitivity of segmentation density to the time threshold, reporting the median number of segments per minute (line) and the interquartile range (shaded). The curve reaches a plateau beyond about 4 seconds, and the selected value of 5 seconds is highlighted by the dashed line.

nounced, while further increases yield diminishing returns. Finally, the Pareto analysis (Fig. 3c) reinforces this conclusion: around $p = 0.2 - 0.3$, the trade-off between coverage and complexity reaches a critical point, providing nearly complete energy retention with significantly reduced complexity compared to lower prominence values. Accordingly, we set $p = .2$ for the remainder of this work since fidelity is the priority.

Once an appropriate prominence value was determined, a second, parallel constraint was incorporated to regularize segment length and prevent overly coarse partitioning in low-dynamic conditions. In fact, while acceleration remains the primary criterion for segmentation, it is not always sufficient across all operational regimes. For instance, when the UAV is hovering, performing steady flight, or remaining on the ground with the motors on, relying solely on acceleration can lead to extended intervals without triggers, producing an overly rough segmentation with excessively long segments. To mitigate this effect, we activate a time-based criterion in parallel: a new segment is forced whenever $\Delta T = 5s$ have elapsed since the previous segmentation trigger, regardless of whether the last trigger was caused by acceleration or by time. To justify the choice of ΔT, we performed a sensitivity analysis of the time threshold τ at fixed prominence ($p = .2$) on the same subset of 50 flights. As shown in Fig. 3d, the number of segments per minute decreases sharply when increasing τ from 1 to about 3 s, indicating that small τ values lead to over-segmentation. Beyond approximately 4 seconds, the curve stabilizes, indicating that segmentation no longer responds significantly to further increases in τ. Accordingly, we selected $\tau = 5$ seconds as a conservative operational threshold within this plateau, to maintain consistent segmentation granularity while minimizing the risk of over-segmentation.

Finally, it is important to note that time-based activations constitute only 1.04 % of all segmentation triggers within the entire training dataset. This indicates that the additional criterion has a limited overall impact, primarily due to the predominance of acceleration-driven triggers. However, it remains a valuable regularization mechanism, especially in enforcing a minimum segmentation density during low-dynamic periods where acceleration alone may not reliably indicate meaningful boundaries. Using this multi-criteria segmentation approach, the flight mission is partitioned into representative segments. For each segment, the average values of all flight variables are calculated, along with the corresponding energy consumption. In particular, the mean current and the energy consumed within each segment are utilized as reference parameters for training energy consumption models.

Let us consider a flight divided into N segments identified by the multi-criteria segmentation procedure. Each segment $n \in \{1, \dots, N\}$ comprises samples indexed by $S_n = \{k_1, k_2, \dots, k_{\mathcal{T}_n}\}$, where $\mathcal{T}_n$ represents the total number of raw data points within the segment. For every sample $k \in S_n$, the log records a wide range of variables including the battery terminal voltage V_k (volts), the battery current I_k (Amperes, defined as positive while discharging), and the elapsed time Δt_k (seconds) between samples $k-1$ and k. The average value of the generic flight variable u for segment n is calculated according to Eq. 2:

$$u_n = \frac{1}{\mathcal{T}_n} \sum_{k \in S_n} u_k. \tag{2}$$

The instantaneous electric power is then $P_k = V_k I_k$, and the energy consumed in the segment n is obtained through the discrete integral:

$$E_n^\star = \sum_{k \in S_n} V_k I_k \Delta t_k, \tag{3}$$

expressed in joules. As already mentioned, the other variables have been calculated by averaging. Eq. 1 reports the calculation of the mean current for the sake of clarity:

$$I_n^\star = \frac{1}{\mathcal{T}_n} \sum_{k \in S_n} I_k. \tag{4}$$

As a result, each segment n is associated with a pair of Ground Truth (GT) labels $(E_n^\star, I_n^\star)$, utilized to train supervised learning models, and a vector $\mathbf{x}_n \in \mathbb{R}^{22}$ containing the other features (Table 2).

Table 2
The 22 input features utilized by the GRU predictors are organized into 12 feature groups.

#	Descriptor
1	Mean wind speed
2	Mean wind azimuth
3	Mean linear velocity components (x, y, z)
4	Mean angular velocity components (roll, pitch, yaw)
5	Mean ground-track speed
6	Payload mass
7	Positional dispersion in x, y, and z
8	Dispersion of smoothed linear acceleration in x, y, and z
9	Dispersion of roll, pitch, and yaw angles
10	Dispersion of the quaternion scalar component
11	Mission-progress metric
12	Segment duration

Fig. 4 shows a graphical workflow that represents the training segmentation strategy used to segment the training data. The system checks the two trigger conditions involving acceleration and time criteria and, when these are met, it identifies breakpoints and calculates the average values between them along with the energy. The peak vector is inherited from the routine presented on top of the figure. The algorithm continues iteratively until the end condition is reached.

The segmentation strategy has led to a mean of 140 segments per flight, with the minimum and maximum number of segments (69, 258) for flight 77 and 258 respectively. As expected, the acceleration criterion is the most triggered one, as the time criterion is triggered only when no meaningful spikes are identified. As already mentioned, the percentage of time triggered criteria is below 2 per cent over the all training dataset while the flight with the highest percentage of time-triggered criteria is flight 272 with almost 8 per cent of time triggers, presenting a quite steady flight trajectory. Fig. 5 shows an example of a segmentation performed for flight 233.

4.2.1.2. Data Preparation. Segments in which the UAV is on the ground or armed but not in flight are removed by specifically excluding samples characterized by low current draw associated solely with onboard electronics rather than active flight propulsion. After applying this criterion, there are no missing entries, making imputation unnecessary.

It is important to clearly differentiate between raw datasets and segmented datasets: the raw dataset consists of original flight recordings at 1 Hz, whereas the segmented dataset comprises summarized statistical features computed for each segment based on multi-criteria triggers. The segmented dataset containing segment vectors $\mathbf{x}_n$ (Table 2) is z-normalized using statistics computed from the entire training partition; the same transformation is consistently applied to the validation and test sets to prevent data leakage. The dataset is split 70 %, 20 %, 10 % into train, validation, and test flights. Splitting is performed by flight with a fixed random seed that guarantees the presence of at least one long-range sortie in the test set.

4.2.2. ML core: segment-level energy and current prediction

Accurate estimation of the electrical load imposed on the battery begins with the now standardized feature vector $\mathbf{x}_n$ containing the averaged values for each flight segment n, identified by the multi-criteria segmentation algorithm described earlier. Table 2 summarizes the twelve feature groups that convey the wind conditions, kinematic states, payload, and dispersion statistics needed to characterize the propulsion demand within each segment.

Reference values for supervised learning originate in the segmentation phase: each segment n is associated with a pair of GT labels $(E_n^\star, I_n^\star)$, which are used to train supervised learning models.

Two separate bidirectional GRU models are developed to estimate segment-level energy and current [36]. GRU models are chosen due to their effectiveness in capturing sequential dependencies and long-term temporal patterns inherent in flight data. Bidirectional GRUs process

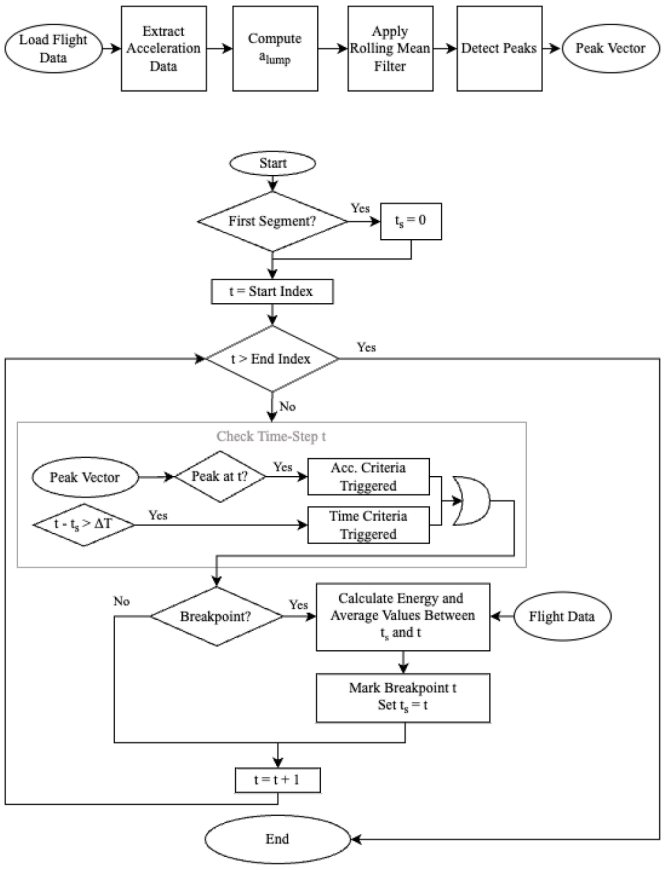


Fig. 4. Offline segmentation strategy employed to segment the training dataset.

data in both forward and backward directions, enabling the model to integrate past and future context to improve prediction accuracy.

The segment-level energy and current prediction models are mathematically defined as:

$$\mathcal{E} : \mathbb{R}^{22} \to \mathbb{R}, \qquad \hat{E}_n = \mathcal{E}(\mathbf{x}_n), \tag{5}$$

$$\mathcal{I} : \mathbb{R}^{22} \to \mathbb{R}, \qquad \hat{I}_n = \mathcal{I}(\mathbf{x}_n), \tag{6}$$

where $\hat{E}_n$ and $\hat{I}_n$ represent the predicted energy consumption and mean current draw for each segment, respectively. These predictions are instrumental in forming the basis for subsequent battery state estimations and voltage predictions in the overall model framework. Training involves iterative optimization of the GRU parameters to minimize the Mean Square Error (MSE) between the predicted values and the GT. Hyperparameters such as hidden size, dropout rate, and the number of layers were systematically explored through random search and validated using a separate validation dataset to ensure optimal generalization performance and robust prediction capability. Table 3 reports the main training and implementation settings of the two GRU predictors used in the proposed framework, with the aim of improving reproducibility and clarifying model comparability. In particular, the table makes explicit the shared training protocol adopted across both predictors and the few architecture-level differences between them. It also clarifies the settings used to support the uncertainty-aware online implementation based on MC dropout.

Table 3
Summary of the main training settings used for the two GRU predictors.

Training setting	Power predictor	Current predictor
Model type	Bidirectional GRU	Bidirectional GRU
Number of input features	22	22
Predicted variable	Segment mean power	Segment mean battery current
Hidden size	32	64
Number of recurrent layers	3	3
Dropout (training)	0.3	0.3
Training sequence length	15 segments	15 segments
Training device	GPU (CUDA)	GPU (CUDA)
Learning rate	1×10^{-4}	1×10^{-4}
Batch size	16	16
Number of epochs	1000	1000

Common preprocessing/training protocol: flight-level split of 70/15/15 (train/validation/test) and z-score normalization computed using training-set statistics only (reported in Section 4.2.1.

4.2.3. Physics-based battery model

UAVs primarily use lithium-based batteries for their power supply, owing to their low self-discharge rate and high energy density [56]. The propulsion battery is represented by a first-order Thévenin equivalent that accurately captures its low-frequency behavior [57]. At each discrete segment n, the terminal voltage is modeled as

$$V_n = V_{\mathrm{oc}}\left(\mathrm{SoE}_n; \boldsymbol{\theta}\right) - I_n R_{\mathrm{int}}, \tag{7}$$

where $\mathrm{SoE}_n \in [0,1]$ is the SOE at segment n, I_n is the segment-average load current (positive during discharge), R_{int} is the lumped internal resistance, and V_{oc} denotes the open-circuit voltage parameterized by the vector $\boldsymbol{\theta} = [\theta_1, \theta_2, \theta_3, \theta_4, \theta_5]^\top$ [58].

The open-circuit voltage is approximated with a five-parameter surrogate

$$V_{\mathrm{oc}}\left(\mathrm{SoE}; \boldsymbol{\theta}\right) = \theta_1 + \theta_2\, e^{\theta_3(\mathrm{SoE}-1)} + \theta_4\,(\mathrm{SoE}-1) + \theta_5\left(e^{-\sqrt{\mathrm{SoE}}} - e^{-1}\right), \tag{8}$$

a functional form that reproduces the plateau-exponential profile observed in lithium-ion cells [59]. All six unknown parameters ($\boldsymbol{\theta}$, R_{int}) are calibrated offline by minimizing the mean-squared error between

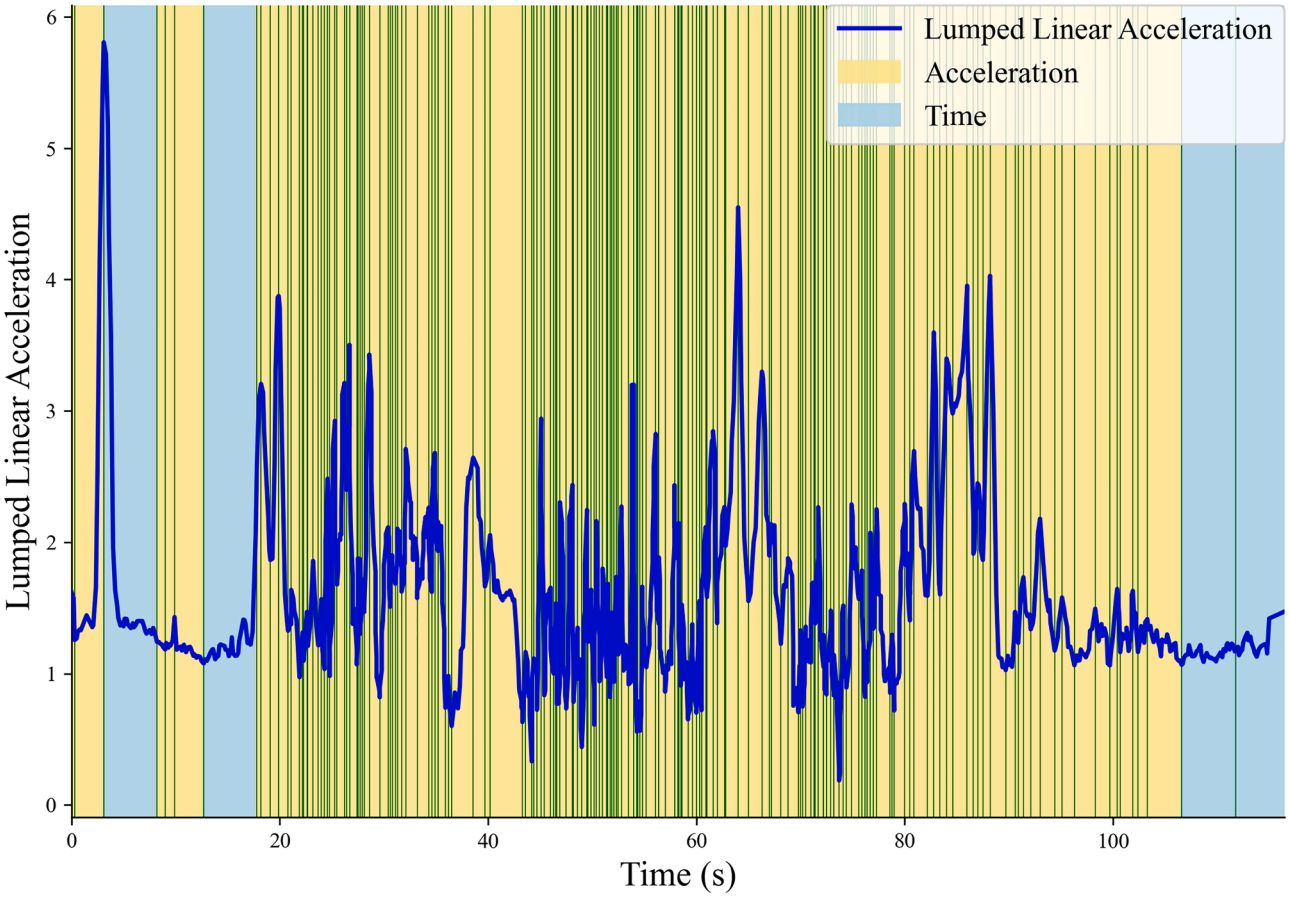


Fig. 5. Segmentation performed for flight 233: 4 segments have been triggered by time criteria in phases where no spikes had been identified.

measured and modeled voltages over a set of representative discharge cycles,

$$\min_{\theta,\, R_{\text{int}}} \frac{1}{N_{\text{cal}}} \sum_{n=1}^{N_{\text{cal}}} \left[V_n^{\text{meas}} - V_{\text{oc}}\left(\text{SoE}_n; \theta\right) + I_n\, R_{\text{int}} \right]^2, \tag{9}$$

using Particle Swarm Optimization (population 50, 200 iterations). Calibration uses the ten training flights that reach the deepest discharges, spanning SoE $\in [0.40, 0.90]$. The SOE evolves segment-wise according to

$$\text{SoE}_n = \text{SoE}_{n-1} - \frac{\hat{E}_n}{E_{\max}} + \eta_n, \tag{10}$$

where $\hat{E}_n$ is the energy consumed during segment n as predicted by the GRU-based model, $E_{\max}$ is the nominal energy capacity of the battery pack, and $\eta_n \sim \mathcal{N}(0, \sigma^2_{\text{SoE}})$ a random variable that represents the uncertainty associated to unmodeled phenomena, such as temperature effects, hysteresis, changes in the internal impedance, among others. When addressing the effects of aging, additional procedures can be implemented to update the parameter $E_{\max}$, which reflects the battery's health condition. For instance, Bustos et al. [60] propose a Bayesian filtering strategy that adapts $E_{\max}$ during operation using voltage and current measurements.

The initial state is sampled as

$$s_0 \sim \mathcal{N}\left(\tilde{s}_0,\, \sigma^2_{\text{S1}}\right),$$

where $\tilde{s}_0$ is the measured initial SoE and σ_{S1} is its measurement noise standard deviation.

Eqs. 7–10 define a stochastic state-space model that is advanced segment by segment using the probabilistic load predictions $\{\hat{E}_n, \hat{I}_n\}$ obtained in Section 4.2.2.

The intrinsic voltage-sensor noise is estimated from stationary segments ($|I| < 0.2$ A for ≥5 s). After detrending with a third-order Butterworth low-pass filter ($f_c = 0.5$ Hz), the RMS value

$$\sigma_V = \text{std}\left(V - V_{\text{trend}}\right), \tag{11}$$

is adopted as the measurement-noise standard deviation, i.e. $\varepsilon_V \sim \mathcal{N}(0, \sigma_V^2)$. For the flights analyzed here, this procedure yields $\sigma_V \approx 115$ mV.

4.2.4. *Battery voltage inference workflow*

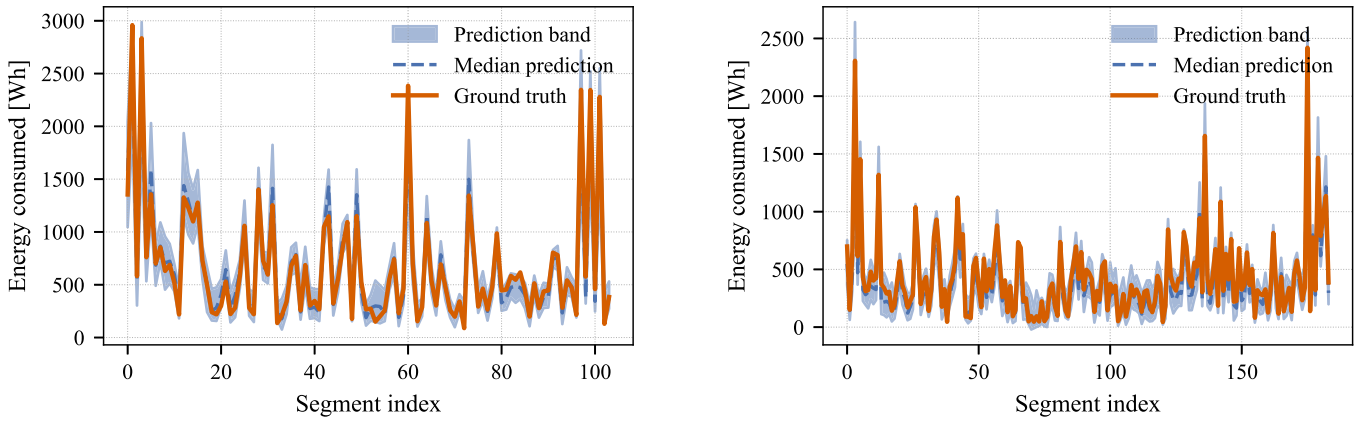
The Algorithm 2 summarizes the inference workflow, which propagates the aleatory and epistemic uncertainties from the GRU outputs to the final energy and voltage estimates. Unless otherwise specified, we decided to draw $M = 33$ MC-dropout samples per segment to balance computational efficiency with accuracy in uncertainty quantification.

Algorithm 2 Uncertainty propagation.

Require: **Measured** initial SoE $\tilde{s}_0$, initial SoE noise variance σ^2_{S1}, process variance σ^2_{SoE}, segments N, MC samples M

```
Initialize s_0 ~ N(s̃_0, σ²_S1)
for n ← 1 to N do
    for m ← 1 to M do
        Sample (Ê_n^(m), Î_n^(m)) using Eq. 3 and 4
        Update SoE_n^(m) via Eq. 10
        Compute V_n^(m) with Eqs. 7 and 8
        Add sensor noise V_n,obs^(m) ← V_n^(m) + ε_V^(m)
Aggregate {SoE_n^(m), V_n,obs^(m)}_{m=1}^M to obtain mean, 5th and 95th percentiles
```

To visualize the results, in the next Section, the MAE on segment energy and voltage will be reported on two test flights.



(a) Flight 164 (b) Flight 233

Fig. 6. Energy-consumption prediction bands and ground-truth comparisons for two representative flights.

4.2.5. ML core results

This subsection assesses how well the trained GRU-battery pipeline predicts segment-level energy demand and terminal voltage across two representative flights.

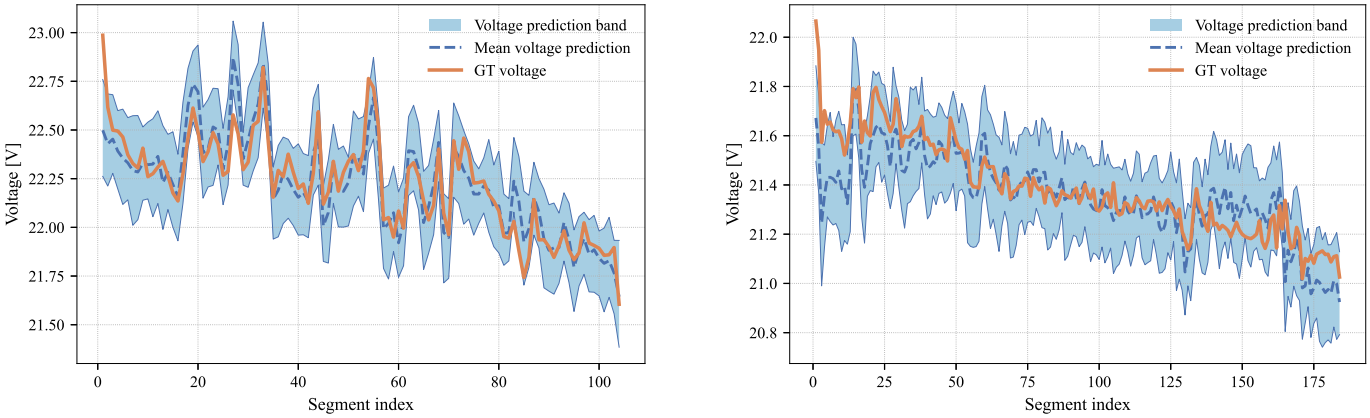
Fig. 6 illustrates the segment-level energy predictions. The 5–95 % band encloses on average 95 % of the ground-truth points, indicating well-calibrated MC-dropout uncertainty. The model tracks both low-power cruise and the short, high-power bursts linked to climbs and rapid manoeuvres: peak loads of 2.9 kW (Flight 164) and 2.7 kW (Flight 233) are reproduced without bias, while low-demand phases remain stable. Segment-level MAE spans 32-47 Wh (1.7-2.8 % of flight totals); cumulative energy bias stays below 3 %. These results confirm that the GRU core learns the combined influence of payload, wind and kinematics, generalizing beyond the training trajectories.

Fig. 7 compares the predicted voltage envelope with the measured terminal voltage. MAE lies between 79 and 100 mV, comfortably below the sensor noise (≈ 115 mV), and 96 % of samples fall within the 5–95 % band. The hybrid GRU-Thévenin layer captures both the initial relaxation and the smooth in-flight decline, yielding a segment-level RMSE of 93 mV without systematic drift.

Taken together, the energy and voltage results show that coupling a data-driven load predictor with a lightweight electrochemical prior yields accurate point estimates and well-quantified uncertainty. Using only 33 MC samples per segment, the framework satisfies real-time onboard constraints and provides confidence metrics suitable for setting adaptive safety margins and supporting mission-planning decisions.

We also assessed alternative sequence-learning architectures for the segment-level load predictor, including bidirectional GRU, bidirectional LSTM, and a lightweight Transformer encoder (TRF). As summarized in Table 4, GRU achieves the lowest MAE and RMSE for both segment energy and mean current. This selection is consistent with the inductive bias of GRUs, which regulate information flow through update and reset mechanisms and often provide stable optimization and strong generalization in small-to-medium data regimes, while remaining computationally efficient for real-time deployment [61]. For these reasons, we adopt GRU as the default model for the ML core in the remainder of this work.

A comparison between the results presented in this article and those obtained in the context of [28] would not be appropriate, as the objectives of the two studies are related but distinct. While Sarkar et al. [28] developed a purely offline energy model with the sole objective of obtaining the lowest possible error, the main objective of this study is to present a comprehensive operational framework, of which the energy model is only one component. As part of a bigger framework, the energy model had to be integrated and connected with the other modules that required segmented trajectories. Although this aggregation slightly reduces the accuracy at individual points, it is instrumental to the integration of the predicted energy demand with the battery-voltage model and to calculate operator-defined risk metrics in real time.



(a) Flight 164 (b) Flight 233

Fig. 7. Voltage-prediction bands and ground-truth comparisons for two representative flights.

Table 4
Comparison of candidate ML architectures for segment-level prediction on the held-out test set. Best (lowest) errors are highlighted in bold.

Variable	Architecture	MAE	RMSE
Energy ($\hat{E}_n$)	GRU	**36.4749**	**53.2711**
Energy ($\hat{E}_n$)	LSTM	40.2654	55.7580
Energy ($\hat{E}_n$)	TRF	41.9413	58.6943
Current ($\hat{I}_n$)	GRU	**0.7677**	**1.0143**
Current ($\hat{I}_n$)	LSTM	0.7694	1.0183
Current ($\hat{I}_n$)	TRF	0.8687	1.2096

5. Framework details

Having established the foundational components of the general framework in Section 3, this section provides a detailed overview of the interactions between the modules and their constituent blocks. As already introduced, the proposed system operates in two phases: an offline planning phase that generates initial voltage predictions from planned trajectories, followed by an online phase that refines these predictions during actual flight operations.

5.1. Offline module

As already illustrated in Fig. 2, the offline module enables users to estimate the voltage discharge trend based solely on planned trajectory information, delivering actionable insights prior to flight execution. This functionality is achieved through the integration of the Route Matching Block and the ML Core introduced in this work.

The primary challenge involves bridging the information gap between the available trajectory data and the input features required by the model. While the flight segmentation protocol detailed in Section 4.2.1.1 relies on acceleration and time variables to generate feature sequences that characterize planned missions, the initial trajectory data only includes positional information. To overcome this limitation, the Route Matching Block generates an approximate acceleration profile by leveraging historical flight data from similar missions. This approach utilizes past flight patterns to reconstruct the missing temporal dynamics essential for accurate mission characterization.

Once the acceleration features are obtained from the Route Matching Block, the complete dataset is segmented according to the established protocol (Section 4.2.1.1). The segmented data is then processed by the ML Core to generate a preliminary voltage trend forecast, providing valuable insights into expected power consumption patterns prior to the start of the mission.

5.1.1. Route matching block

The goal of this module is to generate an a priori acceleration profile of the vehicle. The proposed route matching procedure consists of two sequential steps. In the first step, the coordinates of the target mission are compared with those in the historical database to retrieve the most similar recorded mission. In this work, similarity is measured using DTW. The choice of this metric is based on its effectiveness in comparing sequences with different lengths [62] and invariance to small perturbations [63]. In the second step, the acceleration profile of the most similar recorded mission is transformed to obtain an acceleration profile that matches the length of the target mission. In fact, using DTW, it is possible to recover which indices in the planned route correspond to the ones in the matched route. With this information, the transformed acceleration profile is built by indexing the retrieved acceleration profile to construct a new sequence of accelerations. In case that one index of the target mission is assigned to multiple candidates, this module computes the average value of the assigned accelerations.

The output of this procedure is an inference of the acceleration profile in the route based on the coordinates of the target mission and a historical database of missions. Once the acceleration is computed, the output acceleration can be used to create the segments of the target mission.

In the tested dataset, all of the recorded flights present the same type of operational profile, package delivery. To test the accuracy of the route-matching block, we used the MAE between the reconstructed acceleration profile and the historical acceleration measurements. Indeed, the obtained MAE on the test dataset is 0.48 [$\frac{m}{s^2}$]. As a reference, the mean acceleration value is 3.59 [$\frac{m}{s^2}$], so we consider the obtained match acceptable. Fig. 8 illustrates the obtained reconstruction of the acceleration profiles, in the Z axis, for one of the test flights. From the figure, it is evident that the route-matching block can indeed capture the overall trends in the ground-truth acceleration profiles.

Another sanity check necessary to validate the performance of the route matching block is a sensitivity analysis of the test MAE as a function of the quantity of historical flights in the training. This analysis allows testing the robustness of the route matching block when few historical flights are available. The sensitivity was obtained by randomly generating smaller datasets containing increasing percentages of the available training flights, in increments of 5 %. So, the first dataset contains only 9 historical flights, while the last contains the full 173 training flights.

Fig. 9 shows the accuracy of the route-matching block across different dataset sizes. The results show that the employed route-matching algorithm is robust to changes in the training dataset, as the reported MAE oscillates slightly around the reference MAE. Indeed, for some partitions of the dataset, the MAE becomes even smaller than the reported baseline. This could be due to fewer representative flights, which guide the reconstruction of acceleration and yield slightly more precise trajectories. However, this behavior could indicate overfitting for some particular flights. The obtained result was expected, since all flights represent the same type of mission. For future work, it is proposed to test the route-matching block using data from different mission types and assess its robustness with fewer data samples. In conclusion, we have presented the offline module and the route matching block, which are able to provide a first prediction of the voltage trend given the historical trajectories. We have also performed a sensitivity analysis proving the robustness of the method. Naturally, the accuracy of the offline module prediction is dependent on the similarity with the historical trajectories. If no mission with sufficient similarity is found, the segmentation algorithm will generate feature trajectories that may not accurately represent the current mission. Indeed, because of this possibility, it is necessary to update model predictions if a discrepancy between measurements and predictions is observed during the mission. This is the role of the Online Update Module, presented in the next section.

5.2. Online update module

The online module aims to improve the voltage discharge trend, which is first estimated in the offline phase. The objective is to enable adaptation to unfamiliar mission profiles and those that differ significantly from historical flights. This approach ensures that, even when the evaluated flight deviates notably from past data, the final result is obtained through an online enhancement of the offline trend. This concept relies on the update of the voltage prediction only when a discrepancy is detected between the previous prediction and the actual state. The scheme continuously compares the performance inherited from the offline module with the current system state. In other words, this scheme updates predictions only when the difference between the last forecast and the current system state exceeds a set threshold. This approach avoids unnecessary recalculations of prediction results when the system state has not significantly changed, and enhances the accuracy of the trend through in-flight corrections. It is important to note that the ML model is not retrained but it is just run again with different initial conditions, as an online retraining of the ML model would be both unfeasible and the increase in accuracy would be negligible.

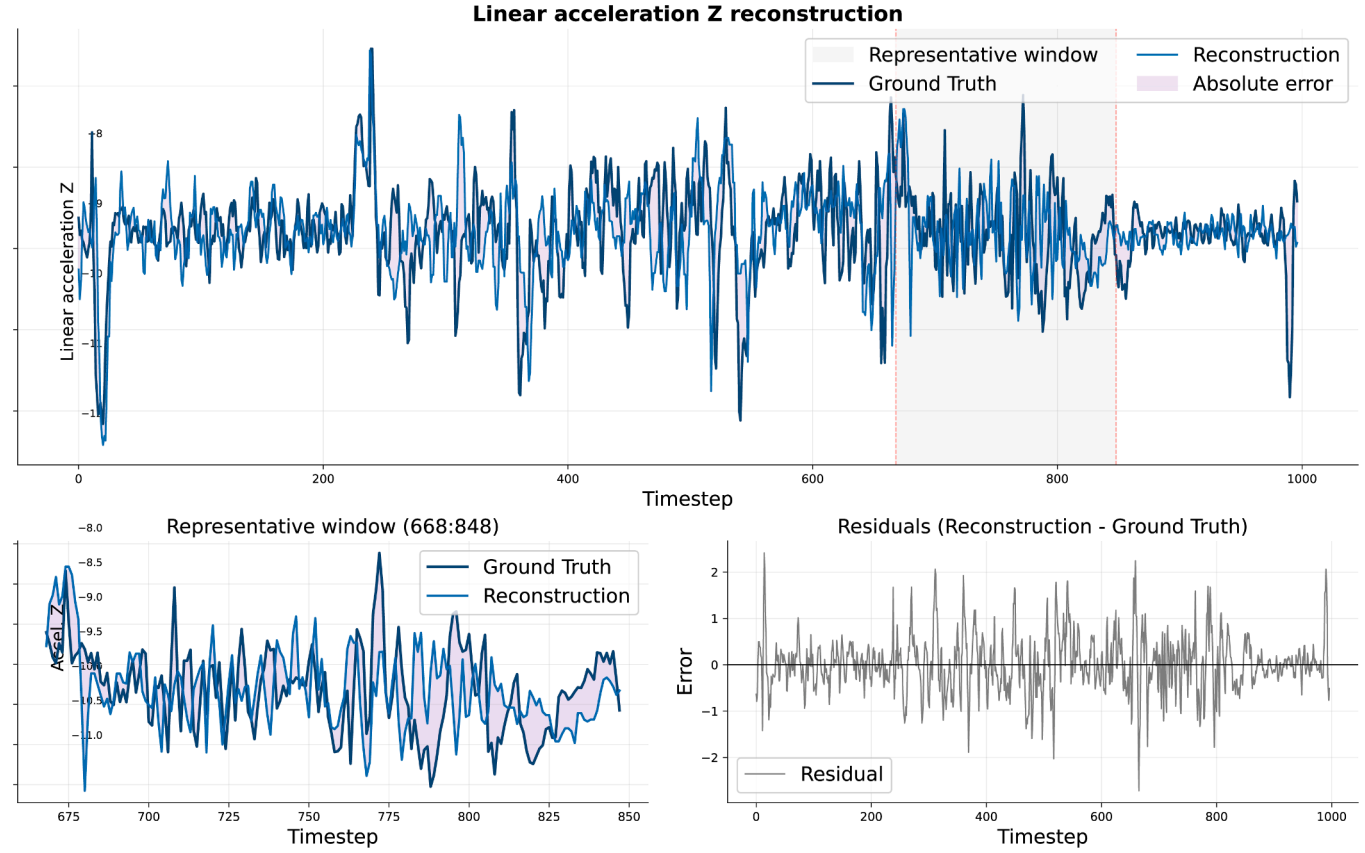


Fig. 8. Ground-truth and reconstructed linear acceleration along Z axis for one of the test flights: full flight, a detail of the window from timestep 668 to 848, and the residual trend.

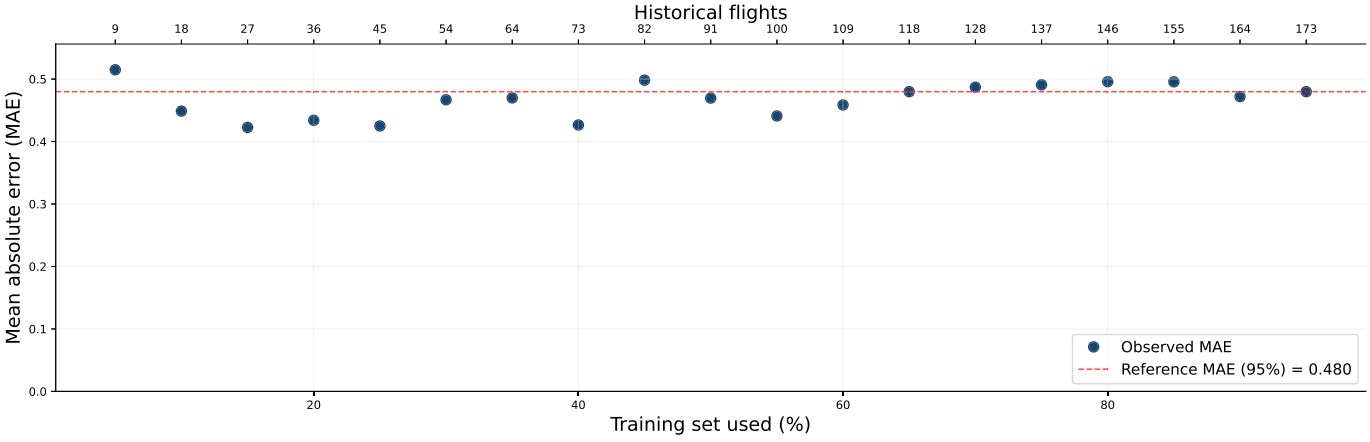


Fig. 9. Observed MAE as a function of training-set size.

5.2.1. Discrepancy-driven prediction metric

The authors have chosen and implemented a discrepancy-driven voltage prediction update scheme, whose initial development was presented in [64]. The key component enabling this selective update is an appropriate online performance metric that measures the similarity between the predicted voltage probability distribution and the voltage GT. If the metric indicates close agreement, the update is skipped; otherwise, the prognostic algorithm runs again using the latest state estimate as the initial condition for long-term predictions. In this case, there is the need for a metric that can effectively compare a deterministic value (i.e., voltage GT) with a probabilistic distribution (i.e., voltage probability distribution). For this scope, among the few available techniques, the CRPS has been selected given the very low computational cost and the high interpretability. CRPS is often the preferred choice to compare the degree to which a probabilistic forecast matches the actual observed outcome and it is expressed through Eq. 12.

$$\mathrm{CRPS}(F, x) = \int_{-\infty}^{\infty} [F(y) - \mathbb{1}\{y \geq x\}]^2 \, dy, \tag{12}$$

where F is the forecast cumulative distribution function, x is the observed value. Finally, the indicator function is highlighted with $\mathbb{1}\cdot$. In the particular case of this work, this formula has been slightly modified to deal with ensemble forecasts (i.e., multiple individual forecasts -

ensemble members) rather than a single prediction, resulting in Eq. 13.

$$\mathrm{CRPS}_{\mathrm{emp}} = \frac{1}{m}\sum_{i=1}^{m}|y_i - x| - \frac{1}{2m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}|y_i - y_j|, \tag{13}$$

where m is the number of ensemble members, y_i is the forecast value of the i-th group, for $i = 1, 2, \ldots, m$, x is the observed (actual) value, $|y_i - x|$ is the absolute deviation of the i-th ensemble member from the observation, $|y_i - y_j|$ is the absolute difference between the i-th and j-th ensemble members. The first term $\frac{1}{m}\sum_{i=1}^{m}|y_i - x|$ represents the MAE between all the ensemble members and the observation, whereas the second term $\frac{1}{2m^2}\sum_{i=1}^{m}\sum_{j=1}^{m}|y_i - y_j|$ represents half the mean absolute difference between all pairs of ensemble members.

The threshold for the prediction update can be customized by the user depending on the preferred refresh rate. To obtain a first estimate, the value corresponding to the 95th percentile of the distribution of all CRPS values has been taken into consideration. In the next Section, a sensibility analysis which can guide the user to select the optimal value of the threshold is presented, closing the loop thus enhancing the framework practical use.

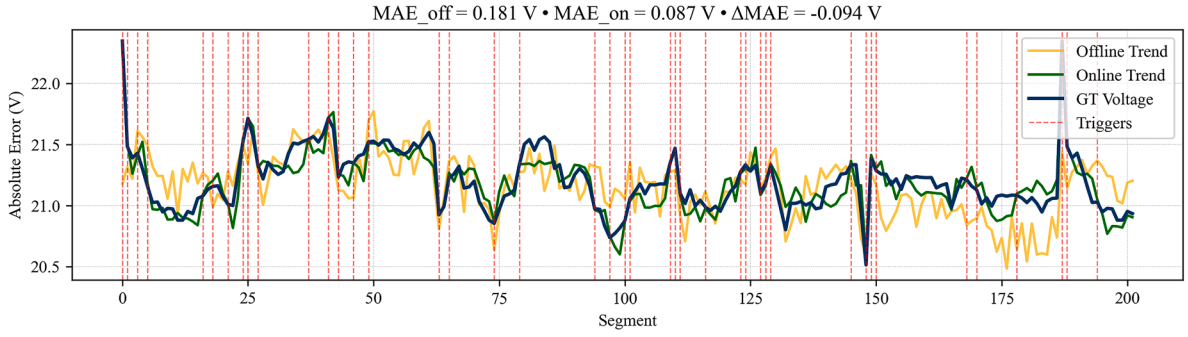
6. Framework results & discussion

The combined use of the offline and online modules effectively manages uncertainty providing the user with a first estimation of the voltage trend and a refined updated trend thanks to the use of the CRPS metric. In the following figures, the results of the framework applied to a brand new Planned Flight PF1 are presented. PF1 data are taken from the testing dataset: the 3D trajectory is used as input, and the flight GT voltage trend is used to evaluate the results. Fig. 10a reports the offline trend, the updated refined trend, along with the voltage GT, while Fig. 10b shows the final trend with the uncertainty band along with the GT for flight 188. The two figures highlight the already satisfactory trend obtained with the sole offline module and the enhancement provided by the online module, which comes into play when the metric exceeds the threshold, resulting in an even better estimate with respect to the GT. The Offline Module achieves a MAE of 0.181 V, which represents satisfactory performance. In contrast, the online module demonstrates significantly superior results with a MAE of 0.087 V, a substantial improvement of 51.9 % over the offline approach. This enhanced performance is attributed to the adaptive trigger mechanism, which was activated 43 times during operation, enabling real-time corrections.

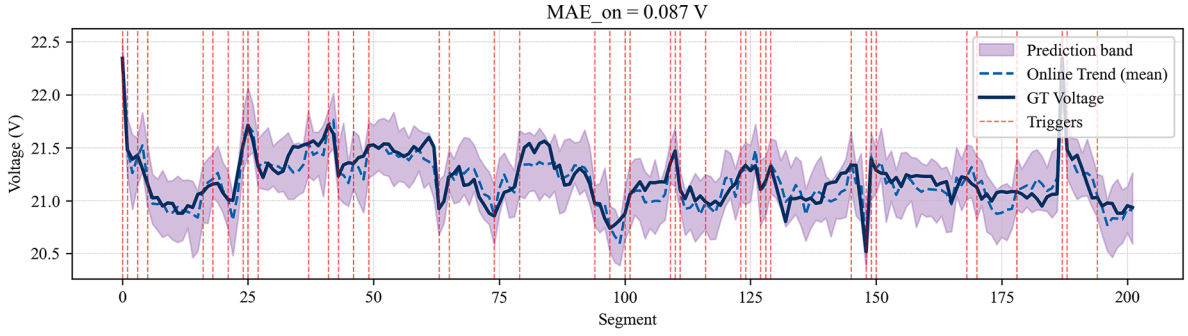
Another perspective is shown in Fig. 11 where each dot represents one flight segment. The value on the x-axis represents the difference between the online and the offline trend: the higher the absolute value, the greater the impact of the online module. The color of the dot underscores the improvement in terms of the error of each segment with respect to the GT. Furthermore, the box-plot comparison of prediction errors relative to GT for flight PF1 reveals a dramatic improvement when using the online module versus the offline approach. The online module significantly reduces the dispersion of errors and effectively constrains the overall magnitude of the error, demonstrating the substantial impact of online processing on the accuracy of the prediction. Finally, it is important to highlight that, as the proposed method forecasts the complete voltage trajectory, the trend can be compared with predefined thresholds associated with battery over-discharge, thereby enhancing mission safety and situational awareness.

6.1. Sensitivity analysis on the CRPS threshold percentile

As a conclusive step, a sensitivity analysis over the selected threshold is shown in Fig. 12. This graph is essential for the practical application of the framework, it further grounds the scienfic base for the selection of the percentile,and it closes the implementation loop, as it assists users in selecting the adjustable threshold parameter.



(a) Comparison between offline, online, and GT voltage trends. The online trend leads to an improvement of more than 51 % (0.094 V) with 43 activations.



(b) Online voltage trend with uncertainty band along with the GT.

Fig. 10. Voltage trends for flight 188. The red dotted lines highlight CRPS triggers where the ML core is run again with updated field voltage data. Note that when the metric triggers an update, the voltage value is corrected using the actual sensor measurement. This explains why the Online Trend aligns perfectly with the GT at each trigger point. (For interpretation of the references to colour in this figure legend, the reader is referred to the web version of this article.)

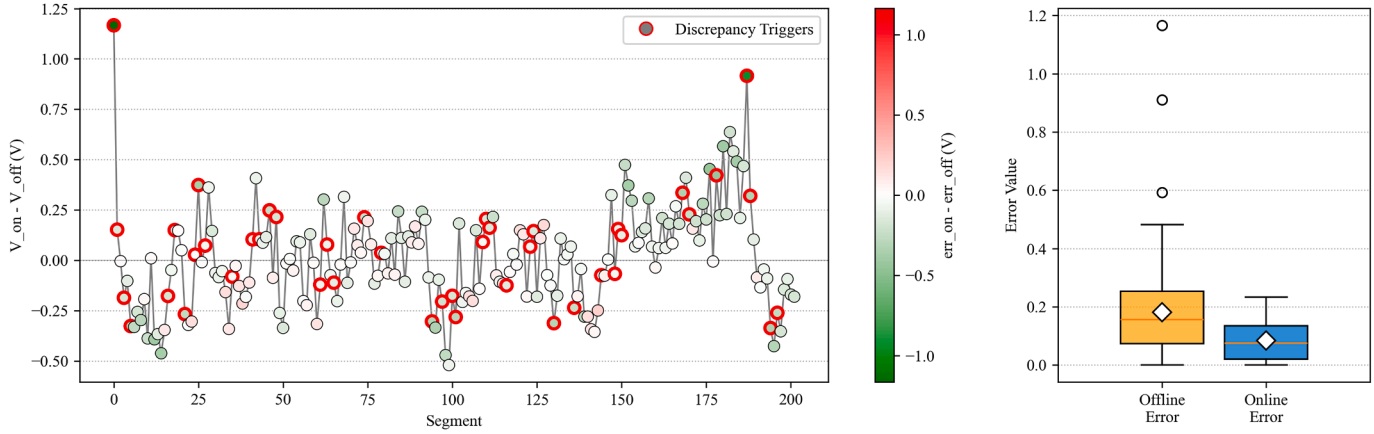


Fig. 11. Impact of the online module on PF1: each dot represents a flight segment. The value of the dots, reported on the y-axis, indicates the difference between the offline and online updated trend. The color of the dots underlines the improvement in terms of the error of each segment with respect to the GT. In other words, the value of the dots represents how much the trend has shifted compared to the offline trend, while the color highlights how the prediction improved. Red circles highlight the activations of the discrepancy triggers.
The side box plot reports online and offline MAEs. The mean is highlighted by the white diamond, while the median is shown with the horizontal orange line.

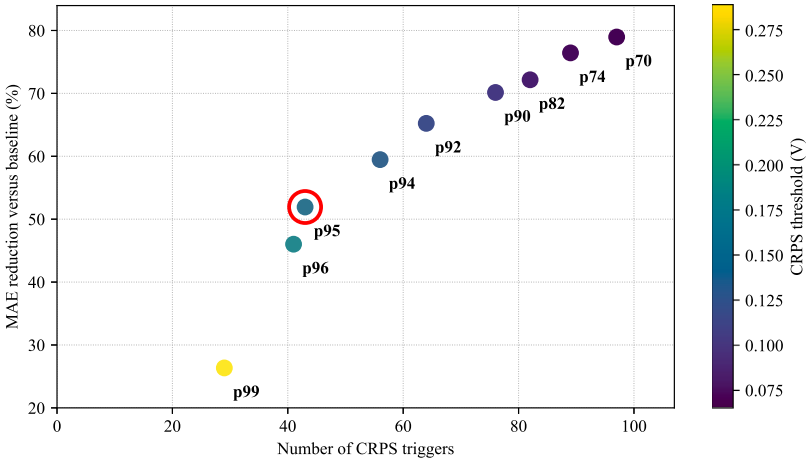


Fig. 12. Sensitivity analysis of the CRPS threshold: trade-off between number of update triggers (computational cost) and MAE reduction versus the offline baseline (PF1).

Thresholds are defined as percentiles of the CRPS distribution and are evaluated in terms of the trade-off between prediction improvement (MAE reduction versus the offline baseline) and computational cost (number of triggered updates). This sensitivity analysis provides the scientific justification for selecting the CRPS threshold, since varying the activation of online updates directly affects both accuracy and computational burden: as expected, lower thresholds increase the trigger frequency and can yield larger MAE reductions, whereas higher thresholds reduce computational load at the expense of smaller gains (Fig. 12). Overall, the online module demonstrates remarkable adaptability across different threshold settings, achieving MAE reductions ranging from 25 % to over 75 % compared to the offline baseline, with each percentile threshold (shown by colored points) representing a different balance point between performance enhancement and trigger frequency.

Based on the observed knee-point behavior, we adopt the 95th percentile (CRPS threshold 0.172 V, highlighted in red) as a balanced operating condition, providing a substantial MAE reduction (from 0.181 V to 0.087 V, 51 % improvement) with a moderate number of triggers. Higher update frequencies can further improve accuracy for applications requiring maximum precision; for instance, a ∼70 % MAE reduction can be achieved using the 90th percentile (CRPS threshold 0.1 V), at the expense of more than 70 triggers. It has to be noted that this choice does not affect the core definition of the proposed framework; rather, it highlights a key strength of the approach, namely its customizability. The proposed visualization and sensitivity study are therefore intended to guide practitioners in selecting an operating point that matches mission requirements and available onboard computational resources.

This visualization enables users to make informed decisions regarding threshold selection and highlights the framework's customizable features.

6.2. *Online computational feasibility and latency–uncertainty trade-off*

To complement the accuracy results and the CRPS-threshold sensitivity analysis, we performed a computational feasibility study by sweeping the number of Monte Carlo (MC) trajectories used during uncertainty propagation. In the proposed implementation, these trajectories are generated from the GRU-based ML predictors by keeping dropout active at inference time (MC dropout), and are then propagated through the physics-based battery model to obtain probabilistic voltage predictions. The benchmark was conducted over the test flights used in the experimental evaluation, and the reported results correspond to averages computed across all test flights for each MC-trajectory configuration. This analysis is intended to assess whether the proposed method can be executed online in practice, and to characterize the trade-off between uncertainty representation fidelity and computational latency. Impor-

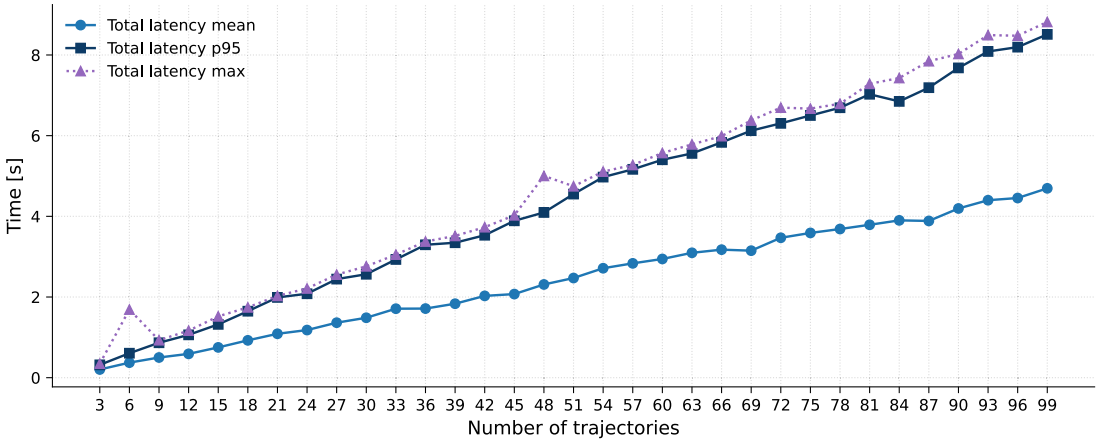


Fig. 13. Latency scaling with the number of MC trajectories. Curves report the mean, 95th percentile, and maximum total computation time measured in the online feasibility benchmark.

Table 5
Summary of online computational feasibility for selected MC trajectory counts, including average time to the next segment, mean block-wise computation times, total mean latency, and the percentage of computations completed before the next segment.

MC trajectories	Average time to next segment [s]	CRPS check mean [ms]	GRU inference mean [s]	Battery model mean [s]	Total latency mean [s]	Completed before next segment [%]
3	0.860	0.025	0.084	0.050	0.054	99.77
6	0.860	0.025	0.213	0.052	0.266	92.09
9	0.860	0.024	0.464	0.060	0.395	84.82
12	0.860	0.025	0.575	0.063	0.538	80.08
15	0.860	0.025	0.768	0.069	0.698	79.09
33	0.860	0.039	1.856	0.082	1.738	52.02

tantly, the present benchmark should be interpreted in the context of online supervisory decision support (mission-level guidance), rather than as a hard real-time inner-loop flight-control requirement.

Fig. 13 shows that the total computation time increases approximately monotonically with the number of trajectories, as expected. In the configuration used in the main experiments ($M = 33$ trajectories), the total latency remains in the order of a few seconds (Table 5), which is compatible with mission-level decision making (e.g., route shortening, early return, or contingency management), where decisions are typically not issued at the same rate as the low-level flight controller. In other words, a latency on the order of a few seconds does not represent a practical limitation for the intended use of the proposed framework.

At the same time, the sweep reveals that latency can be reduced to a practically negligible level (from a mission-time-scale perspective) by decreasing the number of trajectories. In particular, using only 3 trajectories yields sub-second end-to-end latency, as shown in Fig. 13 and summarized in Table 5. This provides a clear operating option when onboard computational resources are limited or when faster refreshes are preferred.

The key point, however, is that the error metrics remain relatively stable across the sweep. As shown in Fig. 14, both MAE and mean CRPS vary within a relatively narrow range compared with the much larger change observed in latency. Therefore, reducing the number of trajectories can substantially decrease computation time while preserving similar predictive performance. Conversely, increasing the number of trajec-

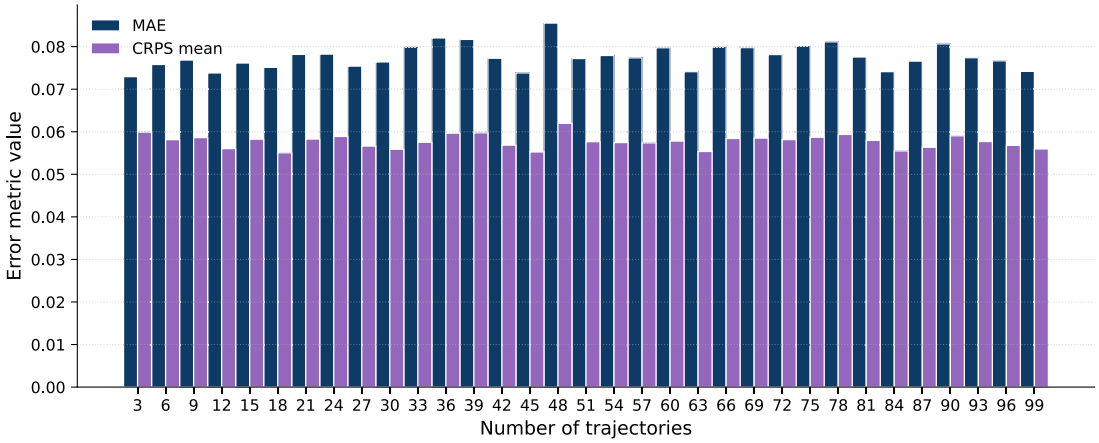


Fig. 14. Prediction error metrics as a function of the number of MC trajectories. Bars report the segment-level MAE and mean CRPS obtained for each configuration in the online feasibility sweep.

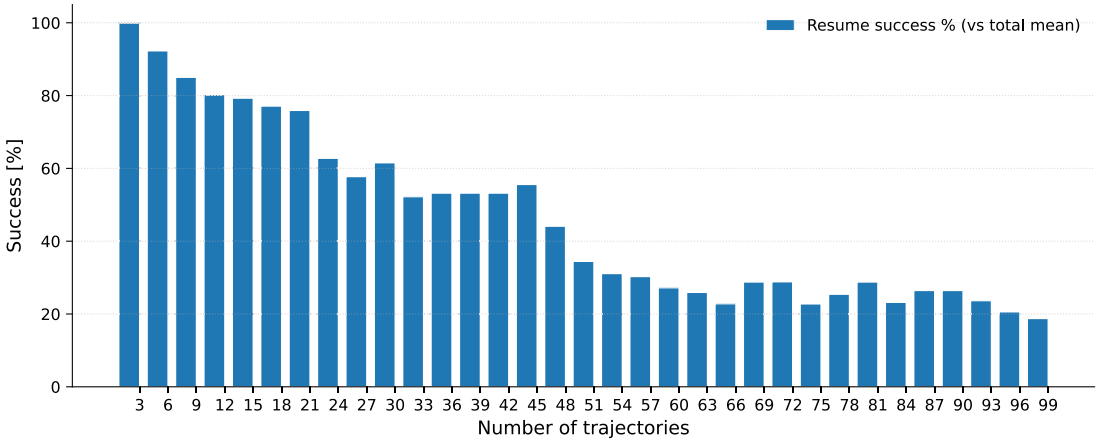


Fig. 15. Online completion rate as a function of the number of MC trajectories. Bars show the percentage of update opportunities for which the full computation is completed before the next segment.

tories improves the MC representation of predictive uncertainty, but at the cost of higher latency. This is the central latency–uncertainty trade-off of the proposed online implementation.

Fig. 15 provides an additional operational perspective by reporting the percentage of update opportunities in which the full computation finishes before the next segment window, using a mean-time budget criterion. As expected, this completion rate decreases as the number of trajectories increases. Even so, the results support the practical feasibility of the framework for online use: the selected $M = 33$ setting offers a reasonable compromise between uncertainty modeling and computational burden, while lower values of M are available when near-immediate updates are prioritized.

Finally, the timing decomposition in Table 5 shows that the discrepancy metric evaluation (CRPS) adds a negligible overhead compared with the GRU inference and battery-model propagation steps. All timing results in this subsection were obtained on a workstation equipped with an AMD Ryzen 9 6900HX processor, 16 GB of RAM, and an NVIDIA GeForce RTX 3060. This is a relevant practical result, since it confirms that the selective update logic is not the computational bottleneck; instead, the dominant cost is associated with propagating the uncertainty ensemble through the predictive pipeline.

6.3. Model generalization to different UAV platforms and operational scenarios

The proposed framework was validated using the dataset presented in Rodrigues et al. [35], which consists of multiple UAVs operating in package-delivery missions. Since the employed prediction models are in part data-driven, they assume that the tested trajectories will also be represented by the training distribution. Thus, if the UAV platform or battery chemistry differs from the training dataset, it would be necessary to fine-tune the model parameters. In this regard, the proposed modular approach has the advantage that the data-driven consumption models presented in Eqs. 5 and 6 are suitable for the application of transfer learning strategies, as shown in Kim et al. [65]. Moreover, for the physics-based battery model, it is possible to adjust model parameters using discharge data collected during operation, as shown in Perez et al. [66]. Thus, by applying these fine-tuning techniques, it would be possible to extend the proposed methodology to other UAV platforms and battery chemistries.

Another implication of this core assumption is that the mission executed during the online stage is similar to at least one of the training trajectories. This means that implementing the proposed framework in cases where the desired mission differs significantly from the reference training set might result in poor performance. Indeed, in this scenario, it would be expected to observe a higher rate of update triggers, implying a general increase in computing time. Moreover, in this configuration, the framework's performance would depend primarily on the online adaptation module, since it allows correcting erroneous associations that may occur in the route-matching block.

7. Conclusions & future developments

The deployment of electric UAVs in urban and mission-critical scenarios is intrinsically linked to the development of reliable and grounded energy and voltage prediction models, which can provide the user with reliable information about the battery SOE. In this context, this paper introduces a dual-phase voltage prediction framework which, given a preliminary planned trajectory, provides the user with an estimation of the voltage trend along the route and then refines this prediction during flight by selectively updating the model only when significant deviations are detected. This novel solution, which employs a probabilistic/deterministic metric, improves the first estimation by rerunning the model only when it is needed, thus limiting the computational burden. By employing a hybrid data-driven and physics-based approach, along with a multi-criteria segmentation strategy for elaborating historical flight data, the framework achieves both accuracy and computational efficiency. The model accuracy, with a MAE around 79 and 100 mV, well within the range of sensor noise, demonstrates its strong ability to capture the complex relationships inherent in UAV flight data. This level of precision not only validates the effectiveness of the model, but also lays a solid foundation for deploying such techniques in real-world operational scenarios. The results underscore the significant benefits of the hybrid Offline-Online approach, evidenced by a reduction in MAE from 0.181 V to 0.087 V (51 % improvement), which highlights how continuous online refinement can substantially enhance prediction performance.

The ablation study on the CRPS threshold parameters offers additional insights, enabling users to intuitively evaluate the trade-offs between error reduction and computational cost, thus supporting more informed decision-making. An additional online feasibility analysis further confirms the practical deployability of the proposed framework for mission-level supervisory decision support. By sweeping the number of MC trajectories generated through MC dropout in the GRU-based predictors, we identified a clear latency–uncertainty trade-off: lower trajectory counts provide near-immediate updates with only minor changes in MAE and mean CRPS, while higher trajectory counts improve the MC representation of predictive uncertainty at the cost of increased latency. Importantly, the timing decomposition shows that the CRPS-based discrepancy check introduces negligible overhead compared with GRU inference and battery-model propagation, confirming that the selective-update logic is not the computational bottleneck.As a result, this system

not only supports flight planning by providing a first estimation of the voltage trend given route peculiarities, but also assists the user during the flight by offering actionable insights into voltage discharge and deviations from initial estimates.

Furthermore, the route-matching procedure leverages a growing dataset of past flights to identify similar routes and refine acceleration trend estimations, ensuring that the accuracy of offline predictions continues to improve as more operational data becomes available. Owing to the modular structure of the proposed framework, several future developments can be naturally incorporated. For example, improving the route-matching stage may further increase prediction accuracy and provide confidence indicators to inform the user about the reliability of the selected historical matches. In addition, more precise strategies can be explored to refine the CRPS-threshold selection, and the framework can be integrated into higher-level mission planning and organizational decision-support schemes. A broader benchmark against alternative uncertainty-quantification and observer-based strategies (e.g., deep ensembles, other probabilistic predictors, and EKF/UKF/PF-based variants) is also an important next step to better position the proposed method in terms of uncertainty fidelity, computational cost, and online deployability. Finally, validation on additional datasets and dedicated live-flight experiments remains a critical step to further assess real-world performance, robustness, and practical deployment beyond offline benchmark data.

In conclusion, in the fastly evolving panorama of UAV research, this work places itself as one of the few ML-powered solutions for voltage prediction and, through its innovative dual-phase architecture, represents an essential build block to enable the integration of battery-powered UAVs into a plethora of applications.

CRediT authorship contribution statement

Leonardo Baldo[†]: Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Jorge E. García Bustos[†]:** Writing – review & editing, Writing – original draft, Visualization, Validation, Software, Methodology, Investigation, Formal analysis, Data curation, Conceptualization; **Benjamin Brito Schiele:** Writing – review & editing, Writing – original draft, Software, Methodology, Formal analysis, Conceptualization; **Ricardo Salas-Espiñeira:** Writing – review & editing, Writing – original draft, Visualization, Software, Methodology; **Andrea De Martin:** Writing – review & editing, Supervision, Funding acquisition; **Marcos E. Orchard:** Writing – review & editing, Supervision, Project administration, Funding acquisition, Conceptualization.

Data availability

The data that support the findings of this study are openly available at https://doi.org/10.1038/s41597-021-00930-x.

Funding

This work was supported in part by ANID FONDECYT 1250036, Advanced Center for Electrical and Electronic Engineering, ANID Basal Project CIA250006. The work of Jorge E. García Bustos has been supported by ANID-PFCHA/Doctorado Nacional/2022-21221213. Leonardo Baldo: This publication is part of the project PNRR-NGEU which has received funding from the MUR - DM 352/2022.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Leonardo Baldo reports financial support was provided by Leonardo Aircraft Division. If there are other authors, they declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

References

[1] S. Jiao, G. Zhang, M. Zhou, G. Li, A comprehensive review of research hotspots on battery management systems for UAVs, IEEE Access 11 (2023) 84636–84650. https://doi.org/10.1109/ACCESS.2023.3301989

[2] G.J.J. Ducard, M. Allenspach, Review of designs and flight control techniques of hybrid and convertible VTOL UAVs, Aerosp. Sci. Technol. 118 (2021) 107035. https://www.sciencedirect.com/science/article/pii/S1270963821005459. https://doi.org/https://doi.org/10.1016/j.ast.2021.107035

[3] F. Tlili, L.C. Fourati, S. Ayed, B. Ouni, Investigation on vulnerabilities, threats and attacks prohibiting UAVs charging and depleting UAVs batteries: assessments & countermeasures, Ad Hoc Netw. 129 (2022) 102805. https://doi.org/10.1016/j.adhoc.2022.102805

[4] A.A. Saadi, B.P. Bhuyan, A. Ramdane-Cherif, Power consumption model for unmanned aerial vehicles using recurrent neural network techniques, Aerosp. Sci. Technol. 157 (2025) 109819. https://www.sciencedirect.com/science/article/pii/S1270963824009489. https://doi.org/https://doi.org/10.1016/j.ast.2024.109819

[5] K. Dorling, J. Heinrichs, G.G. Messier, S. Magierowski, Vehicle routing problems for drone delivery, IEEE Trans. Syst., Man, Cybern.: Systems 47 (1) (2017) 70–85. https://doi.org/10.1109/TSMC.2016.2582745

[6] Q. Cai, Q. Ji, X. Chen, T. Wang, L. Li, Q. Yuan, S. Gao, Y.j. Cheng, Comprehensive study of high-temperature calendar aging on cylinder Li-ion battery, Chem. Eng. Sci. 298 (2024). https://doi.org/https://doi.org/10.1016/j.ces.2024.120355

[7] G. Zhang, X. Wei, G. Han, H. Dai, J. Zhu, X. Wang, X. Tang, J. Ye, Lithium plating on the anode for lithium-ion batteries during long-term low temperature cycling, J. Power Sources 484 (2021). https://doi.org/https://doi.org/10.1016/j.jpowsour.2020.229312

[8] S. Saxena, C. Hendricks, M. Pecht, Cycle life testing and modeling of graphite/Licoo2 cells under different state of charge ranges, J. Power Sources 327 (2016) 394–400. https://doi.org/https://doi.org/10.1016/j.jpowsour.2016.07.057

[9] M.M. Shibl, L.S. Ismail, A.M. Massoud, A machine learning-based battery management system for state-of-charge prediction and state-of-health estimation for unmanned aerial vehicles, J. Energy Storage 66 (2023) 107380. https://doi.org/https://doi.org/10.1016/j.est.2023.107380

[10] W. Tian, L. Liu, X. Zhang, J. Shao, J. Ge, Adaptive hierarchical energy management strategy for fuel cell/battery hybrid electric UAVs, Aerosp. Sci. Technol. 146 (2024) 108938. https://www.sciencedirect.com/science/article/pii/S1270963824000713. https://doi.org/https://doi.org/10.1016/j.ast.2024.108938

[11] R. Alyassi, M. Khonji, A. Karapetyan, S.C.-K. Chau, K. Elbassioni, C.-M. Tseng, Autonomous recharging and flight mission planning for battery-Operated autonomous drones, IEEE Trans. Autom. Sci. Eng. 20 (2) (2023) 1034–1046. https://doi.org/10.1109/TASE.2022.3175565

[12] N.K. Ure, G. Chowdhary, T. Toksoz, J.P. How, M.A. Vavrina, J. Vian, An automated battery management system to enable persistent missions with multiple aerial vehicles, IEEE/ASME Trans. Mechatron. 20 (1) (2015) 275–286. https://doi.org/10.1109/TMECH.2013.2294805

[13] W. Dai, M. Zhang, K.H. Low, Data-efficient modeling for power consumption estimation of quadrotor operations using ensemble learning, Aerosp. Sci. Technol. 144 (2024) 108791. https://www.sciencedirect.com/science/article/pii/S1270963823006879. https://doi.org/https://doi.org/10.1016/j.ast.2023.108791

[14] S. Kodeeswaran, A. Kannabhiran, D. Elangovan, A comparative study of energy sources, docking stations and wireless charging technologies for certain quadrotor unmanned aerial vehicles, Aerosp. Sci. Technol. 166 (2025) 110628. https://www.sciencedirect.com/science/article/pii/S1270963825006996. https://doi.org/https://doi.org/10.1016/j.ast.2025.110628

[15] T.A. Rodrigues, J. Patrikar, A. Choudhry, J. Feldgoise, V. Arcot, A. Gahlaut, S. Lau, B. Moon, B. Wagner, H.S. Matthews, S. Scherer, C. Samaras, In-Flight positional and energy use data set of a DJI matrice 100 quadcopter for small package delivery, Sci. Data 8 (1) (2021) 155. https://doi.org/10.1038/s41597-021-00930-x

[16] M. Elsayed, M. Mohamed, The impact of airspace regulations on unmanned aerial vehicles in last-Mile operation, Transport. Res. Part D: Transport Environ. 87 (2020) 102480. https://doi.org/10.1016/j.trd.2020.102480

[17] J. Zhang, J.F. Campbell, I.I. Donald C. Sweeney, A.C. Hupman, Energy consumption models for delivery drones: a comparison and assessment, Transport. Res. Part D: Transport Environ. 90 (2021) 102668. https://doi.org/10.1016/j.trd.2020.102668

[18] U.C. Cabuk, M. Tosun, O. Dagdeviren, Y. Ozturk, Modeling energy consumption of small drones for swarm missions, IEEE Trans. Intell. Transp. Syst. 25 (8) (2024) 10176–10189. https://doi.org/10.1109/TITS.2024.3350042

[19] P. Beigi, M.S. Rajabi, S. Aghakhani, An overview of drone energy consumption factors and models, in: Handbook of Smart Energy Systems, Springer International Publishing, Cham, 2023, pp. 529–548. https://doi.org/10.1007/978-3-030-97940-9_200

[20] A.M. Moore, Innovative scenarios for modeling intra-city freight delivery, Transport. Res. Interdiscip. Perspect. 3 (2019) 100024. https://doi.org/https://doi.org/10.1016/j.trip.2019.100024

[21] H. Huang, A.V. Savkin, C. Huang, Reliable path planning for drone delivery using a stochastic time-Dependent public transportation network, IEEE Trans. Intell. Transp. Syst. 22 (8) (2021) 4941–4950. https://doi.org/10.1109/TITS.2020.2983491

[22] A. Abdilla, A. Richards, S. Burrow, Power and endurance modelling of battery-powered rotorcraft, in: 2015 IEEE/RSJ International Conference on Intelligent Robots and Systems (IROS), 2015, pp. 675–680. https://doi.org/10.1109/IROS.2015.7353445

[23] H. Gong, B. Huang, B. Jia, H. Dai, Modeling power consumptions for multirotor UAVs, IEEE Trans. Aerosp. Electron. Syst. 59 (6) (2023) 7409–7422. https://doi.org/10.1109/TAES.2023.3288846

[24] H.V. Abeywickrama, B.A. Jayawickrama, Y. He, E. Dutkiewicz, Empirical power consumption model for UAVs, in: 2018 IEEE 88th Vehicular Technology Conference (VTC-Fall), 2018, pp. 1–5. https://doi.org/10.1109/VTCFall.2018.8690666

[25] A.S. Prasetia, R.-J. Wai, Y.-L. Wen, Y.-K. Wang, Mission-Based energy consumption prediction of multirotor UAV, IEEE Access 7 (2019) 33055–33063. https://doi.org/10.1109/ACCESS.2019.2903644

[26] U.C. Çabuk, M. Tosun, R.H. Jacobsen, O. Dagdeviren, A holistic energy model for drones, in: 2020 28th Signal Processing and Communications Applications Conference (SIU), 2020, pp. 1–4. https://doi.org/10.1109/SIU49456.2020.9302218

[27] C. Muli, S. Park, M. Liu, A comparative study on energy consumption models for drones, in: A. González-Vidal, A. Mohamed Abdelgawad, E. Sabir, S. Ziegler, L. Ladid (Eds.), Internet of Things, Springer International Publishing, Cham, 2022, pp. 199–210. https://doi.org/10.1007/978-3-031-20936-9_16

[28] A. Sarkar, F. Santoso, J. Shen, B. Du, A. Telikani, J. Yan, Evaluating energy consumption prediction models of a quadcopter unmanned aerial vehicle, in: 2024 IEEE 100th Vehicular Technology Conference (VTC2024-Fall), 2024, pp. 1–7. https://doi.org/10.1109/VTC2024-Fall63153.2024.10757699

[29] E.I.A. El-Latif, M. El-dosuky, Predicting power consumption of drones using explainable optimized mathematical and machine learning models, J. Supercomput. 81 (5) (2025) 646. https://doi.org/10.1007/s11227-025-07105-0

[30] R. D'Andrea, Guest editorial can drones deliver?, IEEE Trans. Autom. Sci. Eng. 11 (3) (2014) 647–648. https://doi.org/10.1109/TASE.2014.2326952

[31] M.A. Figliozzi, Lifecycle modeling and assessment of unmanned aerial vehicles (drones) CO2e emissions, Transport. Res. Part D: Transport Environ. 57 (2017) 251–261. https://doi.org/https://doi.org/10.1016/j.trd.2017.09.011

[32] J.K. Stolaroff, C. Samaras, E.R. O'Neill, A. Lubers, A.S. Mitchell, D. Ceperley, Energy use and life cycle greenhouse gas emissions of drones for commercial package delivery, Nat. Commun. 9 (1) (2018) 409. https://doi.org/https://doi.org/10.1038/s41467-017-02411-5

[33] T. Kirschstein, Comparison of energy demands of drone-based and ground-based parcel delivery services, Transport. Res. Part D: Transport Environ. 78 (2020) 102209. https://doi.org/https://doi.org/10.1016/j.trd.2019.102209

[34] A. Thibbotuwawa, P. Nielsen, B. Zbigniew, G. Bocewicz, Energy consumption in unmanned aerial vehicles: a review of energy consumption models and their relation to the UAV routing, in: Information Systems Architecture and Technology: Proceedings of 39th International Conference on Information Systems Architecture and Technology – ISAT 2018, Springer International Publishing, Cham, 2019, pp. 173–184. https://doi.org/10.1007/978-3-319-99996-8_16

[35] T.A. Rodrigues, J. Patrikar, A. Choudhry, J. Feldgoise, V. Arcot, A. Gahlaut, S. Lau, B. Moon, B. Wagner, H.S. Matthews, et al., In-flight positional and energy use data set of a DJI matrice 100 quadcopter for small package delivery, Sci. Data 8 (1) (2021) 155. https://doi.org/https://doi.org/10.1038/s41597-021-00930-x

[36] S. Yoon, D. Jang, H. Yoon, T. Park, K. Lee, GRU-Based Deep learning framework for real-Time, accurate, and scalable UAV trajectory prediction, Drones 9 (2025) 142. https://doi.org/10.3390/drones9020142

[37] Z. Lin, Y. Li, Q. Qian, F. Huang, X. Zhang, T. Zhang, W. Wang, Construction of an improved semi-physical simulation system for UAV with integrated energy-consumption prediction model and its evaluation of the path planning algorithms in mountainous scenery, Comput. Ind. Eng. 197 (2024) 110601. https://www.sciencedirect.com/science/article/pii/S0360835224007228. https://doi.org/https://doi.org/10.1016/j.cie.2024.110601

[38] J. Kim, S. Jeon, J. Kim, H. Cha, Voltage prediction of drone battery reflecting internal temperature, in: Proceedings of the 59th ACM/IEEE Design Automation Conference, ACM, 2022, pp. 307–312. https://doi.org/10.1145/3489517.3530448

[39] I.M. Monirul, L. Qiu, R. Ruby, Accurate state of charge estimation for UAV-centric lithium-ion batteries using customized unscented kalman filter, J. Energy Storage 107 (2025) 114955. https://doi.org/10.1016/j.est.2024.114955

[40] S.S.S. Narayanan, S. Thangavel, Terminal voltage prediction of Li-Ion batteries using combined neural network and teaching learning based optimization algorithm, Appl. Soft Comput. 133 (2023) 109954. https://doi.org/10.1016/j.asoc.2022.109954

[41] X. Ding, D. Zhang, J. Cheng, B. Wang, P.C.K. Luk, An improved thevenin model of lithium-ion battery with high accuracy for electric vehicles, Appl. Energy 254 (2019) 113615. https://doi.org/10.1016/j.apenergy.2019.113615

[42] J. Lu, R. Xiong, J. Tian, C. Wang, F. Sun, Deep learning to predict battery voltage behavior after uncertain cycling-induced degradation, J. Power Sources 581 (2023) 233473. https://doi.org/10.1016/j.jpowsour.2023.233473

[43] J. Yang, Y. Chen, Q. Huang, G. Wu, L. Liu, Z. Yang, Y. Huang, Discharge voltage prediction model of batteries in different degradation states based on IWOA-ATCN, Processes 13 (2024) 46. https://doi.org/10.3390/pr13010046

[44] H. Xi, T. Lv, J. Qin, M. Ma, J. Xie, S. Lu, Z. Liu, Prediction of lithium battery voltage and state of charge using multi-Head attention biLSTM neural network, Appl. Sci. 15 (2025) 3011. https://doi.org/10.3390/app15063011

[45] K. Oka, N. Tanibata, H. Takeda, M. Nakayama, S. Noguchi, M. Karasuyama, Y. Fujiwara, T. Miyuki, Deep learning based emulator for predicting voltage behaviour in lithium ion batteries, Sci. Rep. 14 (2024) 28905. https://doi.org/10.1038/s41598-024-80371-9

[46] J.E.G. Bustos, C. Baeza, B.B. Schiele, V. Rivera, B. Masserano, M.E. Orchard, C. Burgos-Mellado, A. Perez, A novel data-driven framework for driving range prognostics in electric vehicles, Eng. Appl. Artif. Intell. 142 (2025) 109925. https://doi.org/https://doi.org/10.1016/j.engappai.2024.109925

[47] J. Zirong, Z. Liang, Z. Zhilong, 3D Trajectory planning of UAV based on DPGA, IEEE Access 9 (2021) 105667–105677. https://doi.org/10.1109/ACCESS.2021.3099836

[48] Federal Aviation Administration, Package Delivery by Drone (Part 135), Technical Report, U.S. Department of Transportation, Federal Aviation Administration, Washington, DC, 2024. Advanced UAS operations guidance. https://www.faa.gov/uas/advanced_operations/package_delivery_drone

[49] F.C.J. Allaire, G. Labonté, M. Tarbouchi, V. Roberge, Recent advances in unmanned aerial vehicles real-time trajectory planning, J. Unmanned Veh. Syst. 7 (4) (2019) 259–295. https://doi.org/10.1139/juvs-2017-0004

[50] A. Bills, S. Sripad, L. Fredericks, M. Guttenberg, D. Charles, E. Frank, V. Viswanathan, A battery dataset for electric vertical takeoff and landing aircraft, Sci. Data 10 (1) (2023) 344. https://doi.org/https://doi.org/10.1038/s41597-023-02180-5

[51] A. Palamas, P. Kolios, Drone onboard multi-modal sensor dataset, 2023, https://doi.org/10.5281/zenodo.7643456

[52] Y. Grigoriou, N. Souli, P. Kardaras, P. Kolios, Drone onboard multi-modal sensor dataset for complex outdoor scenarios, 2024, https://doi.org/10.5281/zenodo.13682870

[53] M. Booker, H. Singh, N. Umasankar, U. Patadia, T. Sudhakar, J. Geng, et al., Data Collected with Package Delivery VTOL Drone, 2023, https://doi.org/10.1184/R1/21997787.v3

[54] K. Góra, P. Smyczyński, M. Kujawiński, G. Granosik, Machine learning in creating energy consumption model for uav, Energies 15 (18) (2022) 6810. https://doi.org/https://doi.org/10.3390/en15186810

[55] H. Gjoreski, M. Gams, Accelerometer data preparation for activity recognition, in: Proceedings of the International Multiconference Information Society, Ljubljana, Slovenia, 1014, 2011, p. 1014.

[56] M.N. Boukoberine, Z. Zhou, M. Benbouzid, A critical review on unmanned aerial vehicles power supply and energy management: solutions, strategies, and prospects, Appl. Energy 255 (2019) 113823. https://doi.org/https://doi.org/10.1016/j.apenergy.2019.113823

[57] C. Burgos-Mellado, M.E. Orchard, M. Kazerani, R. Cárdenas, D. Sáez, Particle-filtering-based estimation of maximum available power state in lithium-Ion batteries, Appl. Energy 161 (2016) 349–363. https://doi.org/https://doi.org/10.1016/j.apenergy.2015.09.092

[58] H. Zhang, M.Y. Chow, On-line PHEV battery hysteresis effect dynamics modeling, in: IECON 2010 - 36th Annual Conference on IEEE Industrial Electronics Society, 2010, pp. 1844–1849. https://doi.org/https://doi.org/10.1109/IECON.2010.5675395

[59] C. Allendes, A. Beltrán, J.E. García, D. Troncoso-Kurtovic, B. Masserano, B.B. Schiele, V. Rivera, F. Jaramillo, M.E. Orchard, J.F. Silva, H. Rozas, S. Rangarajan, Modeling and simulation of thermal effects on electrical behavior in lithium-Ion cells, Annual Conference of the PHM Society 16 (2024). https://doi.org/10.36001/phmconf.2024.v16i1.4080

[60] J.E.G. Bustos, B.B. Schiele, L. Baldo, B. Masserano, F. Jaramillo-Montoya, D. Troncoso-Kurtovic, M.E. Orchard, A. Perez, J.F. Silva, In situ estimation of li-Ion battery state of health using on-Board electrical measurements for electromobility applications, Batteries 11 (2025) 451. https://doi.org/10.3390/batteries11120451

[61] R. Jozefowicz, W. Zaremba, I. Sutskever, An empirical exploration of recurrent network architectures, in: Proceedings of the 32nd International Conference on Machine Learning - Volume 37, ICML'15, JMLR.org, 2015, p. 2342-2350.

[62] Y. Xu, R.D. Clemente, M.C. González, Understanding vehicular routing behavior with location-based service data, EPJ Data Sci. 10 (2021) 12. https://doi.org/10.1140/epjds/s13688-021-00267-w

[63] E. Keogh, C.A. Ratanamahatana, Exact indexing of dynamic time warping, Knowl. Inf. Syst. 7 (2005) 358–386. https://doi.org/10.1007/s10115-004-0154-9

[64] H. Rozas, F. Jaramillo, A. Perez, D. Jimenez, M.E. Orchard, K. Medjaher, A method for the reduction of the computational cost associated with the implementation of particle-filter-based failure prognostic algorithms, Mech. Syst. Signal Process. 135 (2020) 106421. https://doi.org/https://doi.org/10.1016/j.ymssp.2019.106421

[65] D. Kim, Y. Lee, K. Chin, P.J. Mago, H. Cho, J. Zhang, Implementation of a long short-Term memory transfer learning (LSTM-TL)-Based data-Driven model for building energy demand forecasting, Sustainability 15 (2023) 2340. https://doi.org/10.3390/su15032340

[66] A. Perez, F. Jaramillo, C. Baeza, M. Valderrama, V. Quintero, M. Orchard, A particle-Swarm-Optimization-Based approach for the state-of-Charge estimation of an electric vehicle when driven under real conditions, Annual Conference of the PHM Society 13 (2021). https://doi.org/10.36001/phmconf.2021.v13i1.3013