The Aeronautical Journal (2026), **130**, pp. 1546–1565
doi:10.1017/aer.2026.10145

RESEARCH ARTICLE

Mini UAVS' flight data estimation in navigation phase with LSTM method within the CRISP-DM framework

M. Konar[1], D. Özdemir[1,2], M. Fenerci[2,3] and M. Erşen[2,4]

[1]Faculty of Aeronautics and Astronautics, Aeronautical Electrical and Electronics, Erciyes University, Kayseri, Türkiye
[2]Graduate School of Natural and Applied Science, Aeronautical Electrical and Electronics, Erciyes University, Kayseri, Türkiye
[3]Faculty of Computer and Information Technologies, Information Security Technology, Cappadocia University, Nevsehir, Türkiye
[4]Vocational School of Information Technologies, Unmanned Vehicle Department, Kayseri University, Kayseri, Türkiye
Corresponding author: M. Konar; Email: mkonar@erciyes.edu.tr

Received: 8 October 2025; **Revised:** 27 January 2026; **Accepted:** 29 January 2026

Keywords: autonomous flight prediction; flight data estimation; mini UAVS; navigation phase

Abstract

The privatisation of unmanned aerial vehicles (UAVS) used for situational awareness purposes to ensure their own situational awareness based on parameters gives direction to progress and provides a basic framework for future studies. In this context, a unique communication system architecture was proposed for obtaining state-of-the-art mini-UAV data and evaluations were carried out on the basis of data flow and parameters. Within the scope of the evaluation this study postulates a trailblazing approach as a means of optimising flight data pattern recognition by integrating Cross Industry Standard Process for Data Mining (CRISP-DM) and long short-term memory (LSTM)-based predictive depiction. By leveraging the structured framework of CRISP-DM and the sequential learning capabilities of LSTM, this research aims to enhance the accuracy of mini unmanned aerial vehicle systems (UAVS) flight scenario predictive reliability. This framework is applied to navigation phase, where the overall flight trajectory can be seen, and accurate forecasting and pattern recognition are critical for optimising operational efficiency. The findings have displayed the ability to perform high-accuracy predictions of flight parameters within a structured process. The experimental results demonstrate that key flight parameters can be predicted with near–comma-level numerical accuracy, indicating a high level of estimation precision. Through facilitating immediate data transfer and organised navigation phase evaluation, this research provides a methodical strategy for managing flight data, simultaneously contributing notably to UAV decision support systems and self-qualification engineering.

Nomenclature

3D	three-dimensional
ANN	artificial neural networks
CRISP-DM	Cross-Industry Standard Process for Data Mining
GCS	ground control station
GPS	Global Positioning System
GRU	gated recurrent units
LSTM	long short-term memory
MAVLink	micro air vehicle link
MSE	mean square error
OS	operating system
RMSE	root mean square error
RNN	recurrent neural network
SoC	state of charge

© The Author(s), 2026. Published by Cambridge University Press on behalf of Royal Aeronautical Society.

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

TCN	temporal convolutional networks
UAV	unmanned aerial vehicle
UAVS	unmanned aerial vehicle systems
UDP	user diagram protocol

1.0 Introduction

Target-based optimum parameter selection has great importance in UAV operations. In addition to target-based specialised parameter selection, flight data sense-making has a critical role in flight scenario handling and management [19]. There is a critical requirement to appropriate communication system background for the purpose of UAV flight data acquisition. This background enables the exact analysis of flight data and sense-making of flight dynamics. Advanced flight data sense-making may be grounded within situational monitoring and awareness, pattern recognition and relations mapping, adaptive decision-making, predictive analysis and AI integration processes. Over the years industry has been built in a diverse range of facilities and norms about data process management which is appropriate to put forward a standardised framework. Although industry standard model acceptances for data mining processes consist of different numbers of stages, they generally offer relatable sectoral solution steps. CRISP-DM was selected as the industry standard process methodology for data analytics and business intelligence solutions to be performed on the flight data set obtained after the mini-UAV flight due to its comprehensive, state-of-the-art, business and data processes and its standardisation structure accepted throughout the industry [1, 5, 23, 24, 36]. Flight data handling which is vital to ensure effective management of data at every stage of its lifecycle requires taking over all the data in flight scenario. From an overarching standpoint, takeoff and landing phases demonstrate similar patterns, but the navigation phase indicates the flight pattern and scenario common characteristics [42]. As a result of this matter, the navigation phase has been appointed to work on.

In addition to determining the phase and process model used, it is also important to determine the algorithms, methods and models that will be preferred when working with flight data. Reviewing literature in similar scope with flight data, it is seen that studies have been carried out by various methods in many subject headings [2, 7, 12, 17, 19, 22, 31, 40, 44, 47]. It is possible to classify these studies in general terms as data acquisition and communication system studies [12, 19], anomaly studies [44, 47] and decision support studies [2, 7, 16, 17, 31, 40]. Konar et al. designed a UAV for the purpose of obtaining flight data and presented a purpose-oriented communication system solution [19]. Konar et al. studied the similarity-based models for predicting turbofan engine remaining useful life [16]. Borst et al. researched over the remaining useful life prediction of complex systems via introducing convoluted neural networks (CNN)-LSTM network adaptations [3]. Ho et al. present a solution flight data acquisition via wireless sensor network [12]. Yang et al. pointed out a system that provided flight data anomaly diagnosis with the help of deep learning-supported LSTM method [47]. Xiao et al. handled UAV behaviours with a recurrent neural network (RNN)-based anomaly diagnosis method [44]. Dutta et al. used the autonomous UAV decision support system to cope with unwanted foreign plants in agricultural lands through the poor and rich optimisation algorithm [7]. Mihai-Alin et al. have carried out studies on improving the attitude parameters of the hexacopter UAV by means of the finite element method with the control and decision support systems they have proposed [40]. Barabash et al. have built a gradian-based decision support system for UAV control [2]. In the study they carried out, Porat et al. put forward a decision support system in order to identify the UAV with the most appropriate line of sight for the operator's target at a glance [31]. In his study, Keneni created a decision support system through a rule-based artificial intelligence approach to make the decision-making processes transparent and understandable [17]. The prevalent use of artificial intelligence and machine learning methods has been determined in the literature [2, 7, 12, 16, 17, 19, 31 40, 44, 47].

Flight data is evaluated as a time series. Time series are sequential datasets consisting of parameter values recorded at specified sampling intervals over a selected period [49]. When working with real-time data, it is of great importance to address time series problems. Time series analyses are actively

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press



Figure 1. *Mini-UAV flight data sensing interface.*

involved in characterising the data. In recent studies, various deep learning architectures – such as RNN, LSTM networks, gated recurrent units (GRU), temporal convolutional networks (TCN) and transformer models – have been extensively employed to address time series-related problems within the scope of artificial neural network modeling [21, 26, 27, 38, 43]. Following a detailed analysis of alternative neural network structures, it was decided to use the LSTM architecture, which has strong performance in terms of understanding and modeling time series data. To implement the methodology effectively and effectively, the systematic study was carried out by adopting the data mining standard CRISP-DM process model [10, 41, 45, 46, 48].

This research addresses the prevailing gap in the aviation sector, where data mining and machine learning approaches are often unstructured or ad-hoc, by introducing a systematic framework supported by well-defined data management processes. The proposed methodology offers a structured foundation for optimising both business and data workflows, thereby supporting the development of data-driven aviation solutions. In addition, it contributes significantly to self-qualification engineering by enabling the management of highly complex and interrelated datasets. Focusing on the navigation phase – during which the general flight pattern is determined – the framework facilitates detailed phase analysis. Real-time monitoring, enabled through simultaneous data transmission, enhances situational awareness and ensures effective tracking of flight data. These advancements lay the groundwork for future UAV decisions support mechanisms through dynamic evaluation and continuous data integration.

Section 2 describes the background of hardware and software capability to the flight data acquisition process. Section 3 describes the LSTM and CRISP-DM frameworks. Section 4 describes the overall business and data handling process till their evaluation. Section 5 presents the evaluation and deployment processes related to the CRISP-DM process methodology, and Section 6 summarises the results and gives insights into the future UAV decision support studies.

2.0 Flight data sensing process

This section encompasses a mini-UAV interface that allows real-time data management through the coordinated connections between hardware and software subsystems shown in Fig. 1. Information about flight parameters is collected through sensors and transmitted to the data collection port. The port, which works via a wired connection, transmits the information from the sensors to the receiver. The receiver provides transmission to the control transmitter through the telemetry system. The wireless telemetry connection is carried out by means of data communication protocols which are called MAVLink. The remote controller is the main device controlling the UAV. Thanks to the interface connections performed, real-time data tracking is realised.

The real-time communication system is provided by the fact that the transmission between the control receiver and transmitter is in real time. Since this connection is also realised with telemetry support, there is a general expression in the form of telemetry communication. The first phase is the implementation of MAVLink connection, which is the appropriate communication protocol for wireless data transfer. Among the various MAVLink communication types, UDP connection, which is a model with a minimum protocol mechanism, has been preferred. With the establishment of the UDP connection, a serial communication connection is established with ground control station. The suitability of the serial interface's data transmission rate is being verified, and a handshaking mechanism has been integrated into the UDP connection to enhance data security. Once the connection between the flight control card to the ground control station is established, the system waits for the flight control board to send *heartbeat* messages, which are one of the handshaking protocols used by the flight control system. The heartbeat mechanism

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

periodically monitors the viability of the nodes and comments on the real-time data transmission in the system can be performed. MAVLink message encryption will be performed to improve data security on the flight performed for the purpose of obtaining data, preventing attackers from obtaining meaningful information in the appropriate format. The readiness of the communication system for real-time data flow is controlled by means of a heartbeat message. Heartbeat messages contain system and component identifying information. The generated message content contains information about what the intended connection is and what the intended component is. The communication system is constantly monitored through an endless loop. The fact that the receiver intercepted the MAVLink message gives us information about the operational conditions of the mini-UAV. The absence of the message will continue the endless cycle. Thus, it is possible to obtain real-time flight data. Although the operational messages obtained through the infinite loop, the vitality signals of the component and the system via heartbeat and the integrated information are available. The clear identity of the message, which flight parameter the message was sent to, what the parameter sent in the message is and similar issues cannot reveal a clear information. Also, a dictionary has been assigned for the purpose of creating a framework for flight parameter definitions. This dictionary allows us to define which sensor the information comes from, which unit it has and which parameter it is associated with. The data is then converted to string format via the predefined dictionary and hashed together with a timestamp to ensure integrity and traceability. Keeping the data under recording with time information is important in real-time monitoring. The proposed communication system solution has a file instruction approach depending on the UAV energisation condition after performing the airworthiness control of the mini-UAV. File management is provided by parsing the data from the sensors. GPS and battery data are presented with different message headers, thus making attribute extraction easier by determining which flight parameter belongs to which sensor the message contains. Error management is provided by means of error and exception functions integrated into the system, preventing interruption of the data flow. The existence of a continuous debugging mechanism increases the reliability of the data set. During data acquisition, there is a loop that occurs during the general file opening, writing the message to the file, and error management steps. It is desirable to have a continuous data flow and instant access to real-time data.

At this point, the need to keep the system performance at a reliable point is also taken into account. The proposed communication system solution aims to prevent overheating of the system and loss of performance by adding an acceptable time delay to the system. Accordingly, the communication system solution considers it appropriate for the system to go to sleep with 0.01 seconds (10 milliseconds) after each turn in the cycle by means of the internal counter located inside the electronic unit itself. The state of sleep is considered to be the state in which energy is used the least. The proposed communication system model contains the operating system (OS) design within its own structure. MAVProxy, which is the MAVLink Proxy service, is started automatically with the OS command. This automatic start-up facilitates the data flow between the UAV and the GCS. An advantage of the proxy system is that data is cached, and faster access is being provided. The use of a mini-UAV communication system with autonomous launch procedures ensures rapid and consistent commissioning of the system even under difficult operating conditions. As a result of all these critical business steps, a reliable communication architecture is obtained for the mini-UAVS. The mini-UAV shown in Fig. 2 is being prepared for the task of obtaining flight data.

3.0 The methodological framework of mini-UAVS' flight data sense-making

This section outlines methodological background for mini-UAVS' flight data sense making. Methodological background, which is integrated into two main sections, can be examined under comprising LSTM and CRISP-DM.

3.1 LSTM

LSTM networks, a specialised form of RNNs, are designed to effectively capture long-term dependencies by addressing the vanishing gradient problem inherent in traditional RNN architectures. The LSTM

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press



Figure 2. *Related mini-UAVS connection.*

architecture provides a solution to the problem of learning long-term dependencies brought about by the disappearing gradient problem [8, 11, 13, 14, 20, 32].

From an application-oriented perspective, UAV battery-level data collected during the navigation phase exhibit cumulative discharge behaviour, nonlinear temporal trends and long-term dependency characteristics driven by sustained flight dynamics and energy consumption patterns. Capturing these dependencies is critical for accurate battery-level estimation and cannot be sufficiently addressed by shallow learning or short-memory models [6, 9, 18].

Although recurrent neural architectures such as GRUs, autoregressive integrated moving average (ARIMA)-based regressors and traditional machine learning models can handle time-series data to some extent, the specific nature of UAV battery-level dynamics involves long-term dependencies, nonlinear patterns and delayed effects of system states. LSTM networks have been empirically demonstrated to effectively capture such temporal dependencies in energy-related series. For instance, recent work in battery state prediction shows that LSTM models outperform ARIMA and support vector regression (SVR) models on time-series with complex discharge patterns due to their gated memory mechanisms that preserve long-term context [29]. Similarly, in UAV-related time-series settings, LSTM-based approaches have been shown to achieve superior predictive performance compared to shallow models and alternative recurrent architectures. Therefore, guided by both the characteristics of our dataset and comparative findings in the literature, we adopt an LSTM-based forecasting framework for battery-level estimation in UAV navigation-phase data.

Prior to LSTM modeling, the multivariate flight time-series data were restructured using a sliding-window (lookback) approach to explicitly capture temporal dependencies. Each input sequence was constructed by grouping consecutive observations over a fixed lookback window of length L, while the corresponding output represented the battery level at the subsequent time step. This formulation enables the model to learn temporal discharge patterns during the navigation phase.

All input variables were normalised using min–max scaling to the range [0, 1] to ensure numerical stability and to prevent features with larger magnitudes from dominating the learning process. Normalisation parameters were computed exclusively from the training data and subsequently applied to the test data to avoid information leakage.

Following preprocessing, the dataset was reshaped into a three-dimensional tensor of the form (samples, time steps, features), which constitutes the standard input format required for LSTM-based sequential learning.

Taken together, both the intrinsic characteristics of the navigation-phase battery-level time series and the comparative evidence reported in recent literature support the selection of LSTM as a suitable architecture for this study. While RNN architectures struggle to preserve past information due to the vanishing

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

gradient problem, GRU architectures, despite their shorter training times, exhibit performance losses on complex long-term tasks. TCN architectures, although offering the advantage of parallel processing, cannot match LSTM's efficiency in modelling long-term dependencies. Transformers, benefiting from their attention mechanisms, provide a powerful alternative for capturing long-term dependencies but are not always practical due to their high computational costs. In light of these evaluations, LSTM, with its forget gates and advanced architecture, stands out as the preferred choice for complex time series forecasting and other applications where modelling long-term dependencies is critical [10, 41, 45, 46, 48].

3.2 CRISP-DM

For data analytics and business intelligence solutions to be performed on the flight data set obtained after the mini-UAV flight, CRISP-DM is an industry standard process model for data mining [5]. Over the years, the model has developed and reached wide acceptance [34, 37]. When the phases of CRISP-DM methodology are considered, it is necessary to understand the primary target required for context from the perspective of context that the data analyst should apply at the business understanding stage. For the most part, the customer has many goals. The analyst's goal at this stage is to identify important factors that may affect the outcome of the project [28]. At the data understanding stage, the characteristics of the data contained in our data set are revealed. Analyses are carried out to determine the data quality. Whether there is any deficiency in the data or an anomaly is being detected. In this context, various attributes such as the type of properties in the data set and the correlation between the properties are deciphered [4]. The data preparation stage is one of the intensive process-intensive stages in data mining. It is generally assumed that the preparation of the data covers 50–70% of the project time and effort. It may reduce the time to be allocated to show the necessary sensitivity to the stages of business understanding and data understanding performed before this stage [15]. In this phase, the data to be studied are determined, and the selection criteria of the data to be studied are clearly revealed. Data configuration is performed, and data integration is provided. Arrangements regarding the format of the data are carried out at the points deemed necessary [33]. At the modeling stage, the appropriate modeling technique is determined and applied. The aim is to calibrate the parameters to the optimal value. Some of the techniques used when performing this step of the data format can be found on specific requirements. Therefore, you will usually turn back to the data preparation stage and regulations are carried out [25]. During the evaluation phase, the results are controlled in accordance with defined business objectives. Therefore, it is necessary to define further actions and interpretation of the results. Another point to be noted is that the process should be reviewed in general [35]. The implementation phase is associated with the implementation of data mining project results to ensure that they are available and meet the business requirements of end users. The CRISP-DM methodology defines what the application phase requirements are in a clear framework [30].

4.0 Experimental setup of flight data sense-making in mini UAVS' navigation phase

Flight data sense-making solution has been designed through the modeling of statistics and artificial intelligence algorithms, following CRISP-DM model standard for data mining solutions. Experimental setup comprises consecutively over a period of six stages. Table 1 summarises the practice-oriented implementation of the CRISP-DM framework in this study, with particular emphasis on the data preparation and feature selection stage.

4.1 Business understanding

The first step starts with business understanding. To clearly define the business problem, a thorough analysis of the information related to flight data set is crucial. The flight dataset consists of the autonomous flight envelope of the mini-UAV, covering 3 minutes and 35 seconds. Assuming that the flight envelope

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

Table 1. *Practice-oriented CRISP-DM stages*

CRISP-DM phase	What was specifically carried out
Business understanding	Battery-level estimation during the UAV navigation phase was defined as a situational-awareness and decision-support problem related to energy management.
Data understanding	Navigation-phase telemetry consisting of 5,321 time-ordered samples and 10 flight parameters was examined using descriptive statistics and correlation analysis.
Data preparation/ feature selection	Data segments corresponding to the armed (active) flight state were selected. Feature selection was guided by operational relevance to battery behaviour and correlation analysis results, prioritising parameters showing direct or indirect influence on energy consumption. All selected variables were normalised and reshaped into sliding-window sequences for LSTM input.
Modeling	A multivariate LSTM-based time-series forecasting model was implemented to estimate battery level.
Evaluation	Model performance was evaluated using RMSE metrics and predicted-versus-actual comparisons on chronologically separated training and test sets.
Deployment	Results were interpreted in terms of UAV energy monitoring and potential integration into real-time decision-support systems.



Figure 3. *Mini UAV flight trajectory.*

consists of landing, take-off and navigation phases, the rapidly changing parameters during landing and take-off are ignored and the navigation phase, which constitutes the basic characteristic of the flight, is considered for examination purposes. Figure 3 demonstrates an overhead view of the waypoints and planned altitude changes. The envelope selected for the flight data analysis is the part from waypoint number 2 to waypoint number 26.

When the data set was examined, it was observed that it consisted of 12572 data lines. The selected navigation phase of the flight to be examined consists of instantaneous flight data from line 6421 to line 11740. There are 5321 observation units and 10 variables in the data set which is presented via Table 2.

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

Table 2. *Parameters of the mini-UAV flight dataset*

No	Parameter	Description	Unit
1	Received packet count	Packet index captured by the communication system	–
2	Internal counter value	Time-stamp value generated by the flight controller	—
3	GPS latitude	Latitude position of the mini-UAV	degree
4	GPS longitude	Longitude position of the mini-UAV	degree
5	GPS altitude	AGL-referenced vertical altitude	m
6	Ground speed	Resultant ground speed of the mini-UAV	cm/s
7	Battery level	Remaining usable battery capacity	%
8	Battery voltage	Battery terminal voltage	V
9	Battery current	Battery current draw	A
10	Power state (arm)	Mini-UAV power/arm status (0: passive, 128: active)	logical

In order to make general observations about the data set, the data type, head, tail and missing values, quantiles information regarding the data set gives an idea about whether reliable results are produced and the health of the analysis processes.

The flight data set consists of ten flight parameters, namely the number of incoming packets, internal counter value, GPS latitude, GPS longitude, GPS altitude, ground speed, battery level, battery voltage, battery current, and mini-UAV energisation status.

The flight telemetry data were recorded at a fixed sampling frequency determined by the onboard communication and logging system. All parameters were synchronised and logged at the same temporal resolution, ensuring time-aligned multivariate sequences for time-series modelling.

Data collection was conducted under outdoor flight conditions during normal operational scenarios. Environmental factors such as wind and ambient temperature were not explicitly controlled or measured, as the focus of the study was on modeling battery-level dynamics under realistic navigation-phase conditions.

No additional filtering or noise reduction techniques were applied to the raw telemetry data. This decision was made to preserve the original sensor characteristics and to evaluate the robustness of the proposed LSTM-based model under real-world measurement noise.

4.2 Data understanding

In the second stage, which is data understanding, the flight parameters in the data set will be examined in detail. Before working on the data, it is necessary to ensure the integration of the relevant data libraries, which contain specialised data structures and functions, and provide fast and reliable solutions with the aim of creating appropriate conditions for the data to be stored. Then, it is necessary to provide the relevant connections decoupled between the working environment and the data environment. In order to strengthen the understanding of data, the data framework is being created. An examination of the observation units and variables in the flight dataset, where a quick overview of the dataset is provided with the head information in Fig. 4, is a guide during the data understanding phase. It helps us to decipher the data at the points of which variables the dataset consists, what data type expressions are in the columns, what kind of information each property column contains and how this information can be used for analysis, whether there is data consistency, what values the parameters start from, what units of values can be according to the way the values are expressed, whether there are noticeable missing data and whether the data can be categorised among themselves.

The sense-making of the parameters in the data set at the point where it is of great importance to illustrate the relevance of the parameter. Figure 5 correlation heat map of the data set has been given to flight times.

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

	Incoming Packet Count	Internal Counter Value	GPS Latitude	GPS Longitude	GPS Altitude (m)	Groundspeed (cm/s)	Battery Level (%)	Battery Voltage (V)	Battery Current (A)	Mini-UAVS Armed Status
0	6421	1.720875e+09	386541667	355705205	1392.82	11	94	14.762	47.03	128
1	6422	1.720875e+09	386541667	355705205	1392.82	11	94	14.762	47.03	128
2	6423	1.720875e+09	386541667	355705205	1392.82	11	94	14.746	47.03	128
3	6424	1.720875e+09	386541667	355705205	1392.82	11	94	14.746	48.28	128
4	6425	1.720875e+09	386541667	355705205	1392.82	11	94	14.746	48.28	128

Figure 4. *Flight data quick glance with head information.*

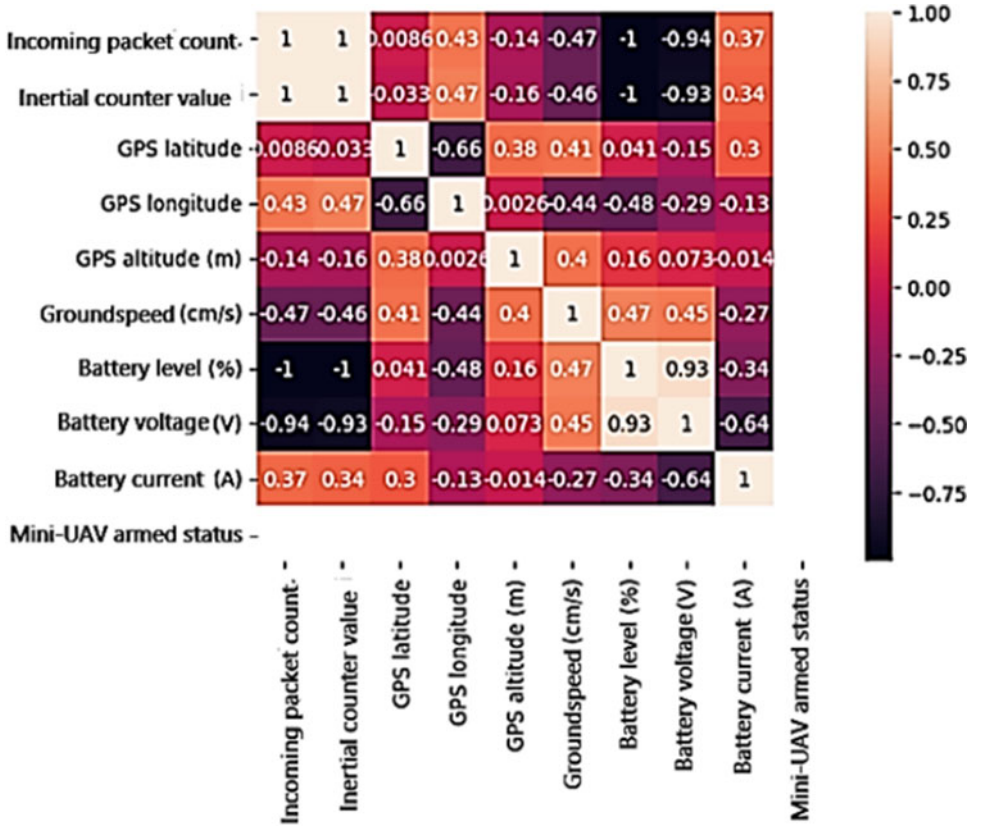


Figure 5. *Mini-UAV flight parameters correlation heat map.*

The starting point of study is data understanding stage which constructs data set sense making and working on it. For this purpose, parameter-based flight data review graphs of the data can be drawn. As a result of the correlation analysis, the battery level that was not assigned to the mini-UAV and showed high relevance to the parameters was selected for examination.

Beyond statistical correlation, battery-level estimation during the navigation phase plays a critical operational role in mini-UAV missions. Navigation-phase flight typically represents the longest continuous energy-consuming segment and directly determines whether the platform can safely complete the

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

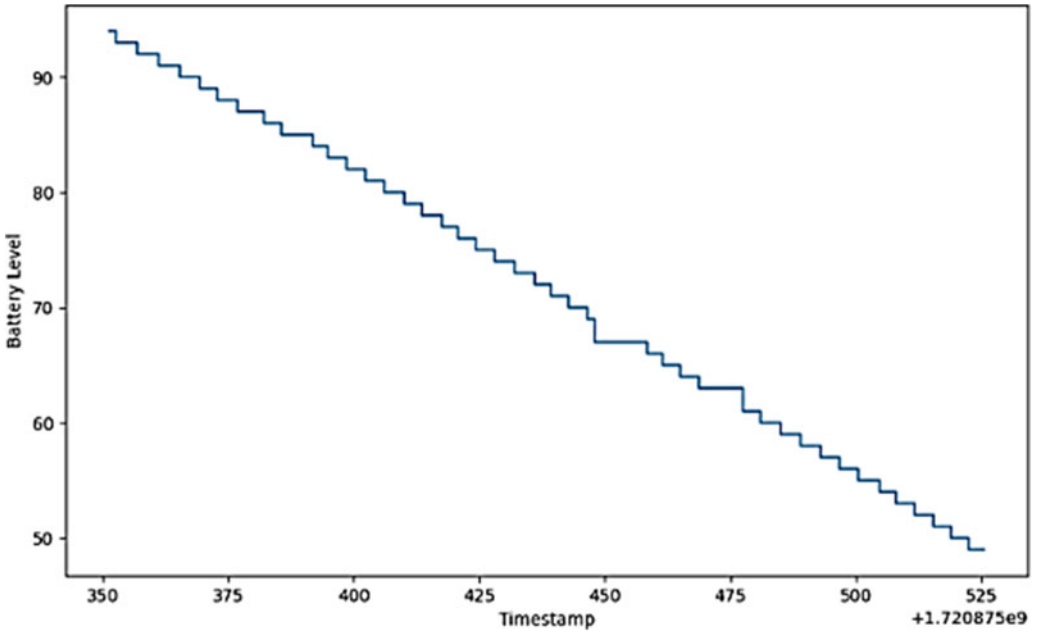


Figure 6. *Battery level over time.*

mission or return to the launch point. Accurate short-term prediction of battery behaviour enhances situational awareness by providing early indications of energy depletion trends, enabling adaptive mission planning such as speed reduction, altitude adjustment or route shortening. Furthermore, battery-level forecasting supports onboard decision-support mechanisms by triggering fail-safe actions and return-to-home logic before critical thresholds are reached. Therefore, predicting battery dynamics during the navigation phase is not only analytically relevant but also essential for operational safety and autonomous mission management in UAV systems. Figure 6 shows the graph of the time-dependent change of the battery level.

4.3 Data preparation

The third step is data preparation which has a prerequisite determination of the parameter to be estimated and the parameters to which it is related. In particular, feature selection during the data preparation stage was driven by both operational relevance to battery dynamics and correlation-based analysis, ensuring transparency in parameter inclusion. When the correlation heat map is examined in order to understand the relationship, parameter selection is performed by taking into account the positive and negative relevance levels on the correlation map. New data frames have been created with internal counter value, GPS longitude, GPS latitude, ground speed, battery voltage, battery current and battery-level parameters. These parameters, which are highly correlated with the battery-level value, are assigned to the index as shown in Fig. 7.

Index-assigned parameters are unprocessed data of the dataset, not used for training or test estimation. This data should be stored as they will create the actual values and show the pattern capture performance in the subsequent process. The corresponding steps are performed to model the change in battery-level values corresponding to the packages assigned to the index shown in Fig. 8.

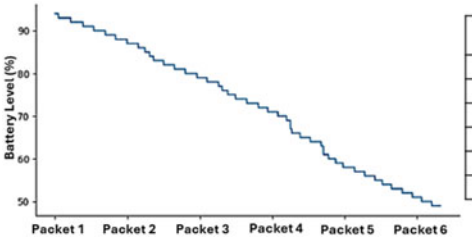
Before the neural network model architecture is created, it is necessary to separate the flight data set into training and test data. At the point of making the training-test distinction, the data set is converted into a training and test format by applying the 80-by-20 distinction, which is accepted as a standard. While the training set helps with the training of the model, the test set functions at the point of evaluating the model performance.

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

timestamp	GPS_lon	GPS_lat	ground_speed	battery_voltage	battery_current	battery_level
1.720875e+09	355705205	386541667	11	14.762	47.03	94
					47.03	94
				14.746	47.03	94
					48.28	94
					48.28	94
...	...	...	...	...	...	...
1.720876e+09	355705349	386542113	16	13.724	52.89	49
					52.89	49
			12	13.724	52.89	49
					52.89	49
				13.772	52.89	49

5320 rows × 1 columns

Figure 7. *Index parameters.*



Timestamp	GPS Longitude	GPS Latitude	Groundspeed	Battery Voltage	Battery Current	Packet Information
1720875351.201045	355705205	386541667	11	14.762	47.03	Packet1
1720875377.026785	355699563	386544219	377	14.48	51.38	Packet2
1720875411.19251	355691799	386549011	301	14.311	50.79	Packet3
1720875440.951564	355695941	386547719	308	13.963	57.31	Packet 4
1720875489.690918	355703999	386544142	400	14.044	48.35	Packet 5
1720875517.507287	355705074	386541986	62	13.758	53.71	Packet 6

Figure 8. *Index parameters with related packets over time.*

To preserve the temporal dependency inherent in time-series flight data, this 80/20 split was performed chronologically, where the first 80% of the sequential observations were used for training and the remaining 20% were reserved for testing. No random shuffling was applied during the splitting process.

Pre-modeling scaling processes are carried out at this stage through data normalisation and standardisation processes. The main reason for scaling is that the variables of methods based on gradient disappearance are of different types from each other, or performance-related problems occur due to the width of the variance, even if it is a single variable. When more than one variable is used, standardisation is also performed in order to prevent the variables from crushing each other.

In this study, all variables were scaled prior to modelling to ensure numerical stability during training and to improve convergence behaviour. Normalisation parameters were computed using only the training data and subsequently applied to the test data to avoid information leakage.

In time series estimates, reshaping for time series data is performed by adding the previous process to the series in size by taking into account the effect of the previous parameter on the next parameter. The data are converted into 3D (three-dimensional) for LSTM estimation.

Specifically, a sliding-window (lookback) approach was employed, where each input sequence was constructed using L consecutive time steps, and the corresponding output represented the battery level at the subsequent time step. Consecutive windows were generated with a stride of one, resulting in overlapping sequences that preserve temporal continuity.

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

```
Model: "sequential"
_________________________________________________________________
 Layer (type)                Output Shape              Param #
=================================================================
 lstm (LSTM)                 (None, 50)                10400

 dropout (Dropout)           (None, 50)                0

 dense (Dense)               (None, 1)                 51

=================================================================
Total params: 10451 (40.82 KB)
Trainable params: 10451 (40.82 KB)
Non-trainable params: 0 (0.00 Byte)
_________________________________________________________________
```

Figure 9. *Model features.*

To sum up, this stage covers all the preprocessing steps before modeling.

Following these preprocessing operations, the structured dataset was reshaped into a 3D tensor of the form (number of samples, lookback window length, number of features), which is required as input for LSTM-based sequential learning.

4.4 Modelling

The fourth step is modelling. As a fundamental prerequisite, it is imperative to first address model features. Since we are working on a time series data set, it is stated that we will produce a sequence model appropriate to the problem type. The LSTM method, which is the appropriate method for the problem, is assigned and the layers that will consist of neurons are determined. With dropout, the model is more generalisable with iterations and fading. The model is shown in Fig. 9. A dropout rate of 0.2 was adopted, as commonly recommended in time-series LSTM applications to balance regularisation and information retention without disrupting temporal dependencies [39, 45].

Figure 9 illustrates the overall LSTM-based forecasting architecture employed in this study. Multivariate flight parameters are first organised into sliding-window sequences and fed into the LSTM layer, where temporal dependencies are learned through gated memory mechanisms. The LSTM output is then passed through a fully connected layer to generate the battery-level prediction for the next time step. Dropout regularisation is applied during training to reduce overfitting and improve generalisation performance.

At this stage, the data set has been brought by customising in accordance with time series analysis. With the process optimisation and evaluation of a set of metrics to determine loss of the model are studied. The model aims to prevent the excess to learn early stopping with the command. It was determined that the model should stop when there is no improvement in validation loss in three periods with the assigned parameters. Training and validation loss related to the changes in the chart in Fig. 10 is presented.

5.0 Results

This section includes stage five and six of CRISP-DM methodology. The fifth step has built into model evaluation. For the purpose of clearly handling model monitoring has been implemented. Figure 10 shows the changes in validation loss and saved models according to the assigned model characteristics.

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

```
Epoch 1/100
210/213 ━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.5107
Epoch 1: val_loss improved from inf to 0.07146, saving model to mymodel.keras
213/213 ━━━━━━━━━━━━━━━━━━━ 4s 5ms/step - loss: 0.5056 - val_loss: 0.0715
Epoch 2/100
195/213 ━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.1156
Epoch 2: val_loss improved from 0.07146 to 0.04243, saving model to mymodel.keras
213/213 ━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.1095 - val_loss: 0.0424
Epoch 3/100
206/213 ━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0589
Epoch 3: val_loss improved from 0.04243 to 0.03564, saving model to mymodel.keras
213/213 ━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0577 - val_loss: 0.0356
Epoch 4/100
200/213 ━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.0478
Epoch 4: val_loss improved from 0.03564 to 0.02715, saving model to mymodel.keras
213/213 ━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0460 - val_loss: 0.0271
Epoch 5/100
208/213 ━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0323
Epoch 5: val_loss improved from 0.02715 to 0.01984, saving model to mymodel.keras
213/213 ━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0318 - val_loss: 0.0198
Epoch 6/100
209/213 ━━━━━━━━━━━━━━━━━━━ 0s 2ms/step - loss: 0.0245
Epoch 6: val_loss improved from 0.01984 to 0.01318, saving model to mymodel.keras
213/213 ━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0242 - val_loss: 0.0132
Epoch 7/100
198/213 ━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.0182
Epoch 7: val_loss improved from 0.01318 to 0.00816, saving model to mymodel.keras
213/213 ━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0175 - val_loss: 0.0082
Epoch 8/100
210/213 ━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.0134
Epoch 8: val_loss improved from 0.00816 to 0.00465, saving model to mymodel.keras
213/213 ━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0133 - val_loss: 0.0047
Epoch 9/100
190/213 ━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.0107
Epoch 9: val_loss improved from 0.00465 to 0.00277, saving model to mymodel.keras
213/213 ━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0101 - val_loss: 0.0028
Epoch 10/100
202/213 ━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.0095
Epoch 10: val_loss improved from 0.00277 to 0.00174, saving model to mymodel.keras
213/213 ━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0093 - val_loss: 0.0017
Epoch 11/100
207/213 ━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.0105
Epoch 11: val_loss improved from 0.00174 to 0.00154, saving model to mymodel.keras
213/213 ━━━━━━━━━━━━━━━━━━━ 2s 5ms/step - loss: 0.0104 - val_loss: 0.0015
Epoch 12/100
202/213 ━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.0108
Epoch 12: val_loss improved from 0.00154 to 0.00146, saving model to mymodel.keras
213/213 ━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - loss: 0.0105 - val_loss: 0.0015
Epoch 13/100
213/213 ━━━━━━━━━━━━━━━━━━━ 0s 4ms/step - loss: 0.0100
Epoch 13: val_loss did not improve from 0.00146
213/213 ━━━━━━━━━━━━━━━━━━━ 1s 5ms/step - loss: 0.0100 - val_loss: 0.0015
Epoch 14/100
213/213 ━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.0103
Epoch 14: val_loss did not improve from 0.00146
213/213 ━━━━━━━━━━━━━━━━━━━ 1s 4ms/step - loss: 0.0103 - val_loss: 0.0016
Epoch 15/100
206/213 ━━━━━━━━━━━━━━━━━━━ 0s 3ms/step - loss: 0.0104
Epoch 15: val_loss did not improve from 0.00146
213/213 ━━━━━━━━━━━━━━━━━━━ 1s 3ms/step - loss: 0.0103 - val_loss: 0.0017
Epoch 15: early stopping
```

Figure 10. *Training and validation loss curves across epochs, demonstrating the convergence behaviour of the LSTM model and the effectiveness of the early stopping mechanism in preventing overfitting.*

Validation and early stopping have been assigned the reference to the change in losses determined in accordance with the mechanism that came into effect. In Fig. 11 The Post scaling performed in the range of 0–1 loss is a graph of values. Figure 12 shows that step by step progress over the tess loss indications. Loss of MSE (mean square Error) are quoted. The retraining processes for the model evaluation phase are given in Fig. 13.

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

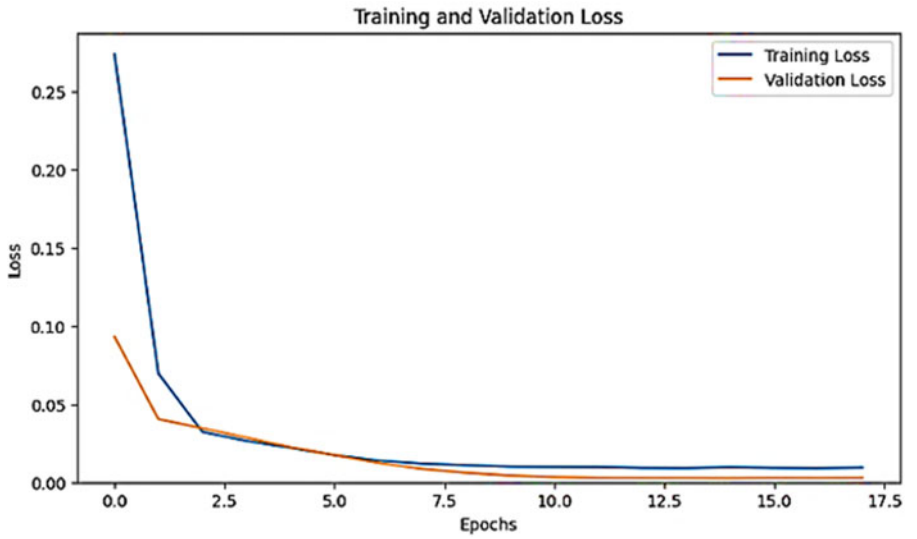


Figure 11. *Normalised loss variation over epochs after post-scaling within the 0–1 range, illustrating the stability and convergence trend of the training process.*

```
34/34 [==============================] - 0s 3ms/step - loss: 0.0028

Test loss:0.3%
```

Figure 12. *Test loss analysis indicating the prediction error behaviour of the trained LSTM model on unseen data samples.*

```
133/133 [==============================] - 1s 2ms/step
34/34 [==============================] - 0s 2ms/step
```

Figure 13. *Training progress of the LSTM model, showing the learning dynamics and optimisation behaviour throughout the training iterations.*

```
Train RMSE: 1.664233922958374
Test RMSE: 0.42267873883247375
```

Figure 14. *Comparison of RMSE values for training and test datasets, highlighting the predictive accuracy and generalisation capability of the proposed model.*

The root mean square error (RMSE) value obtained as a result of the training processes is symbolised by bringing it to the same format as the original data unit. Figure 14 shows the RMSE value for training and test data.

The data obtained as a result of training, and the data related to the real battery level are shown in Fig. 15. It compares the predicted and actual battery level during the training phase, demonstrating the model's ability to closely track the observed battery dynamics. It will be observed that there is a deviation in the prediction performance at the rate specified in the train RMSE value. Figure 16 shows the connection between the data obtained as a result of training and the real values. The graph presents

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

timestamp	GPS_lon	GPS_lat	ground_speed	battery_voltage	battery_current	battery_level	prediction
1.720875e+09	355705205	386541667	11	14.762	47.03	94	95.444656
				14.746	47.03	94	95.444656
					48.28	94	95.444656
					48.28	94	95.444656
					48.28	94	95.444656

Figure 15. *Difference between actual and predicted battery-level values in the training dataset, illustrating the estimation deviation and fitting accuracy of the LSTM model.*

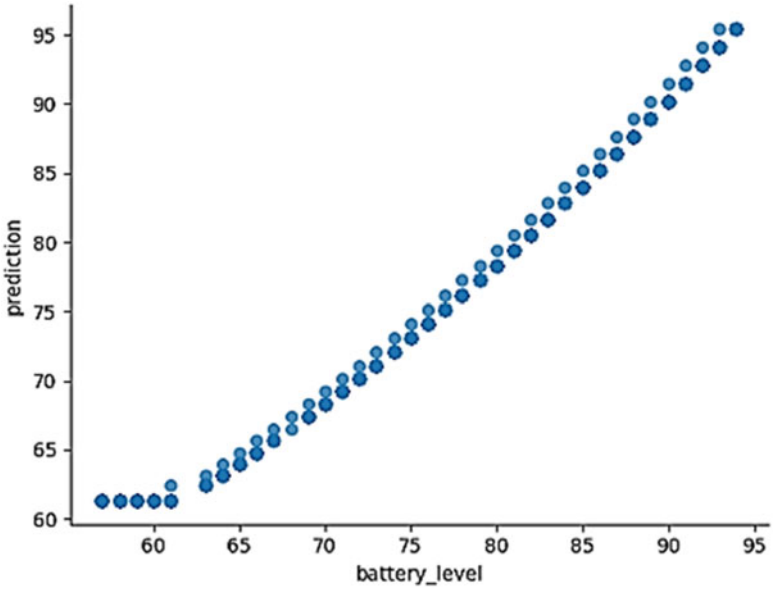


Figure 16. *Relationship between predicted and actual battery-level values in the test dataset, demonstrating the correlation and consistency of the model predictions.*

the predicted versus actual battery level on the test dataset, indicating the generalisation capability of the proposed LSTM model.

The data obtained as a result of the test and the data related to the real battery level are shown in Fig. 17. It illustrates the temporal evolution of prediction error, highlighting stable model behaviour across the navigation phase.

It will be observed that there is a deviation in the estimation performance at the rate specified in the test RMSE value. Figure 18 shows the connection between the data obtained as a result of the test and the real values. The graph summarises the overall prediction performance, showing consistent alignment between estimated and measured battery levels.

Figures 19 and 20 show the pattern-capturing performance of training and test values against the actual values.

CRISP-DM sixth stage of deployment phase is associated with the implementation of data mining project results to ensure that they are available and meet the business requirements of end users. Lifelike and high-precision capture, pattern, performance and applicability confirms.

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

timestamp	GPS_lon	GPS_lat	ground_speed	battery_voltage	battery_current	battery_level	prediction
1.720875e+09	355704557	386541934	299	13.905	52.05	57	57.312355
		386541844	299	13.905	52.05	57	57.312355
	355704532	386541844	299	13.905	52.05	57	57.312355
					52.05	57	57.312355
			240	13.905	52.05	57	57.312355

Figure 17. *Comparison of real battery-level measurements and LSTM-based prediction results over time, illustrating the temporal tracking performance of the model.*

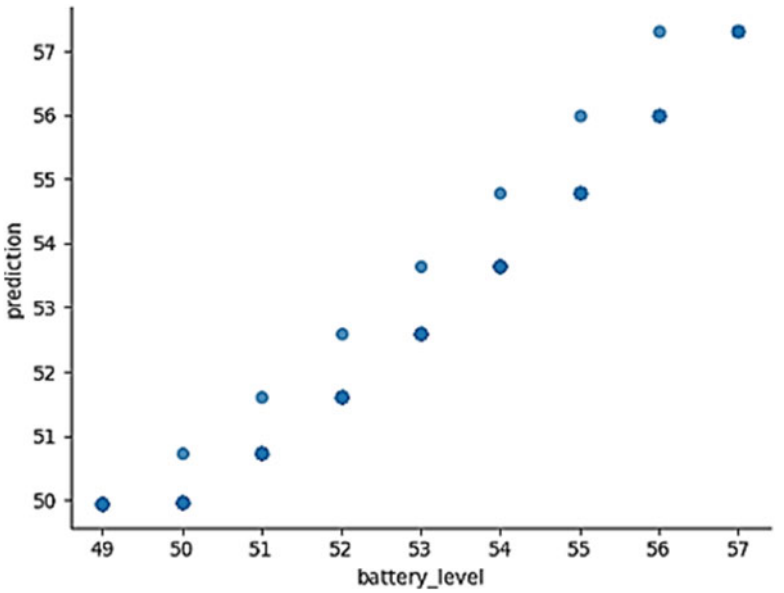
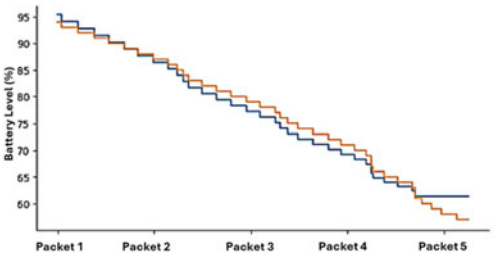


Figure 18. *Scatter plot of predicted versus actual battery-level values, highlighting the distribution of prediction accuracy and potential deviation patterns.*

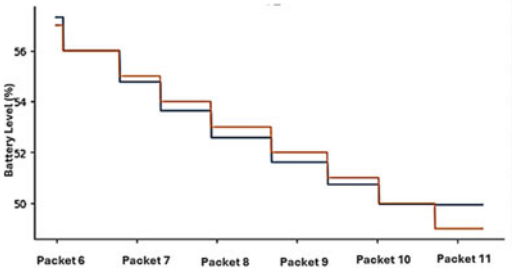


Timestamp	GPS Longitude	GPS Latitude	Groundspeed	Battery Voltage	Battery Current	Packet Information
1720875351.276168	355705205	386541667	11	14.762	47.03	Packet1
1720875377.026785	355699563	386544219	377	14.489	51.38	Packet2
1720575411.19251	355691808	386549011	301	14.311	50.79	Packet3
1720875440.951564	355695941	386547719	308	13.963	57.31	Packet 4
1720875489.690918	355703999	386544142	398	14.044	48.35	Packet 5

Figure 19. *Training set LSTM estimation and real values.*

The obtained results demonstrate that the proposed LSTM-based model is capable of capturing the temporal dynamics of battery-level variation during the UAV navigation phase, as reflected by the low RMSE values and the close alignment between predicted and actual battery trajectories. The model effectively learns nonlinear patterns and delayed effects arising from multivariate flight parameters, indicating its suitability for navigation-phase energy monitoring.

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press



Timestamp	GPS Longitude	GPS Latitude	Groundspeed	Battery Voltage	Battery Current	Packet Information
1720875496. 437752	355704557	386541934	299	13.905	52.05	Packet 6
172087502.012855	355703913	386541926	147	13.776	56.7	Packet 7
1720875508.438525	35570455	386542048	32	13.968	47.44	Packet 8
1720875513.688919	355704749	386542024	83	13.817	51.9	Packet 9
1720875518.944174	355705172	386541992	42	13.753	53.54	Packet 10
1720875523.944124	35570532	386542089	26	13.779	52.98	Packet 11

Figure 20. *Test set LSTM estimation and real values.*

Despite these strengths, the current approach has certain limitations. The model focuses on single-parameter prediction (battery level), and the size of the dataset is limited to a single flight scenario, which may restrict generalisability. In addition, only an LSTM architecture was evaluated in this study. Alternative deep learning models such as GRU, TCN or transformer-based architectures may offer advantages in terms of computational efficiency, parallel processing or long-range dependency modeling. A comparative evaluation of these architectures is therefore identified as a promising direction for future work.

6.0 Conclusion

In today's world, especially at the point of increasing situational awareness, UAVs are actively involved. Mini UAVs stand out with their features such as lightness, compactness and being the most suitable for obtaining information in terms of effective mission performance in difficult mission conditions. In this study, a general analysis of flight parameters obtained as a result of obtaining flight data by mini-UAV flight was performed, the predictability of the parameters was evaluated and the relationships of the parameters according to flight scenarios were analysed by revealing the sections taken from the parameter set in the study. This study can be considered as a concrete example of situational monitoring of UAVs [29].

Although the proposed LSTM-based framework demonstrates strong predictive performance for the analysed flight dataset, several limitations should be acknowledged. First, the model was trained on a single mini-UAV flight dataset, which may restrict its generalisation capability under different environmental conditions, flight profiles or UAV configurations. Second, LSTM networks require relatively high computational resources and training time, which may limit their direct deployment on embedded or real-time onboard systems without optimisation. Finally, the model performance is sensitive to data quality and sensor configurations, indicating that additional calibration and retraining may be required when adapting the framework to different UAV platforms.

In terms of scalability, the proposed CRISP-DM-based framework provides a flexible structure that can be adapted to different UAV platforms. By redefining sensor mappings, updating data preprocessing pipelines and retraining or fine-tuning the LSTM model with platform-specific datasets, the methodology can be extended to multi-platform UAV applications. Future studies may also investigate transfer learning strategies and lightweight neural architectures to enable efficient deployment on resource-constrained UAV systems.

Despite the promising results obtained in this study, several limitations should be acknowledged. The proposed framework was evaluated using flight data collected from a single mini-UAV platform and under specific navigation-phase conditions, which may limit the direct generalisability of the findings to different UAV configurations, mission profiles or environmental conditions. In addition, the LSTM model was trained using a fixed set of input parameters and a predefined lookback window, without adaptive optimisation of temporal context length.

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

Beyond its predictive performance, this study makes several distinctive contributions to the existing literature. First, it demonstrates the structured integration of the CRISP-DM methodology with an LSTM-based time-series estimation model for UAV navigation-phase battery monitoring, providing a transparent and reproducible analytical workflow. Second, the study is supported by a real-time mini-UAV flight data acquisition infrastructure, enabling the use of authentic flight telemetry rather than simulated or synthetic datasets. This combination of systematic data mining principles with real-world UAV operational data represents a practical and methodological advancement over prior studies that primarily focus on standalone modeling approaches. Accordingly, the proposed framework not only contributes to battery-level estimation accuracy but also offers a transferable methodological template for data-driven UAV flight analysis.

Future work will focus on extending the proposed framework by incorporating data from multiple UAV platforms and mission scenarios, exploring adaptive windowing strategies and comparing LSTM-based forecasting with alternative architectures such as GRU and transformer-based models. Furthermore, integrating real-time battery prediction into onboard decision-support systems is expected to enhance operational robustness and autonomy in UAV applications.

Acknowledgments. This study was supported by the Scientific Research Projects Unit of Erciyes University with the FYL-2023-12576 project code. Thank you for your support.

Data availability statement. All data used during the study are available from the corresponding author by request.

Refe rences

[1] Azevedo, A. and Santos, M.F. KDD, SEMMA and CRISP-DM: A Parallel Overview, IADS-DM, 2008, p 152.
[2] Barabash, O.V., Dakhno, N. and Shevchenko, H. "Modified gradient method in a decision support system for control unmanned aerial vehicles," science and education a new dimension, *Nat. Tech. Sci.*, 2017, **16**, pp 60–62.
[3] Borst, N. and Verhagen, W.J.C. Introducing CNN-LSTM network adaptations to improve remaining useful life prediction of complex systems, *Aeronaut. J.*, 2023, **127**, (1318), pp 2143–2153.
[4] Cazacu, M. and Titan, E. Adapting CRISP-DM for social sciences, *BRAIN: Broad Research in Artificial Intelligence and Neuroscience*, 2021, **11**, Suppl. 2, pp 99–106.
[5] Chapman, P., Clinton, J., Kerber, R., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R. The CRISP-DM User Guide, 4th CRISP-DM SIG Workshop, Brussels, Mar. 1999.
[6] Duan, Y., Lv, Y. and Wang, F.Y. Travel time prediction with LSTM neural network, *2016 IEEE 19th International Conference on Intelligent Transportation Systems (ITSC), IEEE*, Nov. 2016, pp 1053–1058.
[7] Dutta, A.K., Albagory, Y., Sait, A.R.W. and Keshta, I.M. Autonomous unmanned aerial vehicles based decision support system for weed management, *Comput. Mater. Contin.*, 2022, **73**, (1), pp 899–915.
[8] Gao, R., Huo, Y., Bao, S., Tang, Y., Antic, S.L. and Epstein, E.S. Distanced LSTM: time-distanced gates in long short-term memory models for lung cancer detection, in D. Shen, T. Liu, Y. Shi, and X. Xu (Eds.), *Machine Learning in Medical Imaging, MLMI 2019*, Lecture Notes in Computer Science, Vol. 11861, Springer, 2019, pp 310–318.
[9] Gers, F.A., Schmidhuber, J. and Cummins, F. *Continual* prediction using *LSTM with* forget gates, in B. Apolloni, M. Marinaro, and R. Tagliaferri (Eds.), *Neural Nets WIRN Vietri-99*, Springer, 1999, pp 133–138.
[10] Gopali, S., Abri, F., Siami-Namini, S. and Namin, A.S. December. A comparison of TCN and LSTM models in detecting anomalies in time series data. *2021 IEEE International Conference on Big Data (Big Data), IEEE*, 2021, pp 2415–2420.
[11] Graves, A. Long short-term memory, in *Supervised Sequence Labelling with Recurrent Neural Networks*, Springer, 2012, pp 37–45.
[12] Ho, D.T., Grøtli, E.I., Sujit, P.B., Johansen, T.A. and Sousa, J.B., Optimization of wireless sensor network and UAV data acquisition, *J. Intell. Rob. Syst.*, 2015, **78**, pp 159–179.
[13] Hochreiter, S. and Schmidhuber, J. Long short-term memory, *Neural. Comput.*, 1997, **9**, (8), pp 1735–1780.
[14] Hu, J., Wang, X., Zhang, Y., Zhang, D., Zhang, M. and Xue, J. Time series prediction method based on variant LSTM recurrent neural network, *Neural. Process. Lett.*, 2020, **52**, pp 1485–1500.
[15] IBM, IBM SPSS Modeler CRISP-DM Guide, IBM Corp., 2002. [Online] Available: https://www.ibm.com/docs/it/SS3RA7_18.3.0/pdf/ModelerCRISPDM.pdf [accessed Jul. 2024].
[16] Isbilen, F., Bektas, O. and Konar, M. Deep learning and similarity-based models for predicting turbofan engine remaining useful life: insights from the CMAPSS dataset, *Aeronaut. J.*, 2025, **129**, (1337), pp 2004–2035.
[17] Keneni, B.M. Evolving Rule Based Explainable Artificial Intelligence for Decision Support System of Unmanned Aerial Vehicles, M.S. Thesis, University of Toledo, OH, 2018.

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

[18] Khalil, K., Eldash, O., Kumar, A. and Bayoumi, M. Economic LSTM approach for recurrent neural networks, *IEEE Trans. Circuits. Syst. II Express. Briefs.*, 2019, **66**, (11), pp 1885–1889.
[19] Konar, M., Özdemir, D., Erşen, M. and Fenerci, M. Acquisition of flight data from mini unmanned aerial systems, *J. Aviat.*, 2024, **8**, (3), pp 221–228.
[20] LeCun, Y., Bottou, L., Bengio, Y. and Haffner, P. Gradient-based learning applied to document recognition, *Proc. IEEE*, 1998, **86**, (11), pp 2278–2324.
[21] Li, D. Preprocessing and analysis method of unplanned event data for flight attendants based on CNN-GRU, *Int. J. Adv. Comput. Sci. Appl.*, 2024, **15**, (11), pp 899–915.
[22] Luan, Y. and Lin, S. Research on text classification based on CNN and LSTM, *2019 IEEE International Conference on Artificial Intelligence and Computer Applications (ICAICA), IEEE*, 2019, pp 352–355.
[23] Mariscal, G., Marban, O. and Fernandez, C. A survey of data mining and knowledge discovery process models and methodologies, *Knowl. Eng. Rev.*, 2010, **25**, (2), pp 137–166.
[24] Matignon, R. *Data Mining Using SAS Enterprise Miner*, John Wiley & Sons, 2007, New York.
[25] Nadali, A., Kakhky, E.N. and Nosratabadi, H.E. Evaluating the success level of data mining projects based on CRISP-DM methodology by a fuzzy expert system, *2011 3rd International Conference on Electronics Computer Technology, IEEE*, Apr. 2011, 6, pp 161–165.
[26] Nanduri, A. and Sherry, L. Anomaly detection in aircraft data using recurrent neural networks (RNN), *2016 Integrated Communications Navigation and Surveillance (ICNS), IEEE*, Apr. 2016, pp 5C2–1.
[27] Nanyonga, A., Wasswa, H. and Wild, G. Phase of flight classification in aviation safety using LSTM, GRU, and BiLSTM: a case study with ASN dataset, *2023 International Conference on High Performance Big Data and Intelligent Systems (HDIS), IEEE*, Dec. 2023, pp 24–28.
[28] NCR, P.C., Clinton, J., NCR, R.K., Khabaza, T., Reinartz, T., Shearer, C. and Wirth, R., CRISP-DM 1.0, NCR Corp., 1999.
[29] Özdemir, D. Acquisition of flight data of mini unmanned aerial vehicles (Master's thesis, Erciyes University, Institute of Science, Kayseri, Turkey), 2024.
[30] Plotnikova, V., Dumas, M. and Milani, F. Adapting the CRISP-DM data mining process: a case study in the financial services domain, *International Conference on Research Challenges in Information Science*, Springer, May 2021, pp 55–71.
[31] Porat, T., Oron-Gilad, T., Silbiger, J. and Rottem-Hovev, M. 'Castling Rays': A Decision Support Tool for UAV-Switching Tasks, CHI'10 Extended Abstracts on Human Factors in Computing Systems, 2010, pp 3589–3594.
[32] Pulver, A. and Lyu, S. LSTM with working memory, *2017 International Joint Conference on Neural Networks (IJCNN), IEEE*, May 2017, pp 845–851.
[33] Saltz, J.S. CRISP-DM for data science: strengths, weaknesses and potential next steps, *2021 IEEE International Conference on Big Data (Big Data)*, IEEE, Dec. 2021, pp 2337–2344.
[34] Schäfer, F., Zeiselmair, C., Becker, J. and Otten, H. Synthesizing CRISP-DM and quality management: a data mining approach for production processes, *2018 IEEE International Conference on Technology Management, Operations and Decisions (ICTMOD), IEEE*, Nov. 2018, pp 190–195.
[35] Schröer, C., Kruse, F. and Gómez, J.M. A systematic literature review on applying CRISP-DM process model, *Procedia. Comput. Sci.*, 2021, **181**, pp 526–534.
[36] Shafique, U. and Qaiser, H. A comparative study of data mining process models (KDD, CRISP-DM and SEMMA), *Int. J. Innov. Scientific. Res.*, 2014, **12**, (1), pp 217–222.
[37] Shearer, C. First CRISP-DM 2.0 Workshop Held, [Online], Available: https://www.kdnuggets.com/news/2006/n19/4i.html [accessed Aug. 2024].
[38] Shi, Z., Xu, M., Pan, Q., Yan, B. and Zhang, H. LSTM-based flight trajectory prediction, *2018 International Joint Conference on Neural Networks (IJCNN), IEEE*, Jul. 2018, pp 1–8.
[39] Srivastava, N., Hinton, G., Krizhevsky, A., Sutskever, I., and Salakhutdinov, R. Dropout: a simple way to prevent neural networks from overfitting, *J. Mach. Learn. Res.*, 2014, **15**, (1), pp 1929–1958.
[40] Stamate, M.A., Pupăză, C., Nicolescu, F.A. and Moldoveanu, C.E. Improvement of hexacopter UAVs attitude parameters employing control and decision support systems, *Sensors*, 2023, **23**, (3), p 1446.
[41] Tuli, S., Casale, G. and Jennings, N.R. Tranad: deep transformer networks for anomaly detection in multivariate time series data. arXiv preprint arXiv:2201.07284, 2022.
[42] Wenzel, K.E., Masselli, A. and Zell, A. Automatic take off, tracking and landing of a miniature UAV on a moving carrier vehicle, *J. Intell. Rob. Syst.*, 2011, **61**, pp 221–238.
[43] Wu, T., Lin, Z., Huang, J., Ding, Y., Maihemuti, A. and Xu, X. Flight quality assessment in full flight phase based on KOA-CNN-GRU-self-attention, *Bull. Pol. Acad. Sci. Tech. Sci.*, 2024, **72**, (6), e151677.
[44] Xiao, K., Zhao, J., He, Y., Li, C. and Cheng, W. Abnormal behavior detection scheme of UAV using recurrent neural networks, *IEEE Access.*, 2019, **7**, pp 110293–110305.
[45] Yadav, A., Jha, C.K. and Sharan, A. Optimizing LSTM for time series prediction in Indian stock market, *Procedia. Comput. Sci.*, 2020, **167**, pp 2091–2100.
[46] Yamak, P.T., Yujian, L. and Gadosey, P.K. December. A comparison between ARIMA, LSTM, and gru for time series forecasting, *Proceedings of the 2019 2nd International Conference on Algorithms, Computing and Artificial Intelligence*, 2019, pp 49–55.

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press

[47] Yang, L., Li, S., Zhang, Y., Zhu, C. and Liao, Z. Deep learning-assisted unmanned aerial vehicle flight data anomaly detection: a review, *IEEE Sens. J.*, 2024, **24**, (20), pp 31681–31695.
[48] Zhang, J. and Man, K.F. October. Time series prediction using RNN in multidimension embedding phase space, *SMC'98 Conference Proceedings. 1998 IEEE International Conference on Systems, Man, and Cybernetics (Cat. no. 98CH36218), IEEE*, 1998, 2, pp 1868–1873.
[49] Zhang, J. and Zhang, P. *Time Series Analysis Methods and Applications for Flight Data*, Springer, 2017, Berlin/Heidelberg, Germany.

Cite this article: Konar M., Özdemir D., Fenerci M. and Erşen M. (2026). Mini UAVS' flight data estimation in navigation phase with LSTM method within the CRISP-DM framework. *The Aeronautical Journal*, **130**, 1546–1565. https://doi.org/10.1017/aer.2026.10145

https://doi.org/10.1017/aer.2026.10145 Published online by Cambridge University Press