

Review

Flight Data-Driven LSTM-Family Models for Resource-Aware Edge Deployment in Aerial Systems: A Review and Methodological Evaluation

Fang Wang ¹, Tianjing Liu ^{1,*}, Yongzheng Wang ², Yixin Zhang ¹, Zhe Wei ³ and Hang He ⁴

¹ School of Science, Civil Aviation Flight University of China, Guanghan 618307, China; wf@cafuc.edu.cn (F.W.); zhangyixin0214019@cafuc.edu.cn (Y.Z.)

² Flight Test College, Civil Aviation Flight University of China, Guanghan 618307, China; wangyongzheng@cafuc.edu.cn

³ School of Computer Science and Artificial Intelligence, Civil Aviation Flight University of China, Guanghan 618307, China; findz@cafuc.edu.cn

⁴ School of Economics and Management, Civil Aviation Flight University of China, Guanghan 618307, China; hehang@cafuc.edu.cn

* Correspondence: liutianjing@cafuc.edu.cn

Abstract

Flight data-driven modeling has become an important approach for trajectory prediction, anomaly detection, and risk assessment in unmanned aerial vehicles and other aerial systems. Such data are usually high-dimensional, nonlinear, multirate, and non-stationary, especially during maneuvering flight, environmental disturbance, and mission-phase transitions. Traditional physics-based methods and shallow machine learning models often have limited adaptability in these conditions, while Long Short-Term Memory (LSTM) networks and their variants have shown strong potential for learning temporal dependencies from complex flight sequences. This paper reviews the development and application of LSTM-family models for flight data analysis, with attention to both methodological performance and resource-aware deployment. The reviewed models include basic LSTM, BiLSTM, CNN-LSTM, ConvLSTM, LSTM autoencoder, attention-enhanced LSTM, graph-based LSTM, and uncertainty-aware LSTM. Their applications are discussed in three main areas: flight trajectory prediction, anomaly detection, and risk assessment. Beyond prediction accuracy, this review also examines robustness to distribution shift, physical consistency, interpretability of anomalies, uncertainty estimation, and onboard implementation. A four-layer and ten-dimensional evaluation framework is presented across data characteristics, model performance, physical-mechanism consistency, and system-engineering constraints. The evidence shows that model size, runtime memory, computational cost, inference latency, energy consumption, and target hardware are often insufficiently reported, while direct quantitative onboard validation remains scarce. This gap highlights the need for standardized deployment reporting.



Academic Editor: Manohar Das

Received: 18 June 2026

Revised: 24 July 2026

Accepted: 31 July 2026

Published: 3 August 2026

Copyright: © 2026 by the authors.

Licensee MDPI, Basel, Switzerland.

This article is an open access article distributed under the terms and conditions of the [Creative Commons Attribution \(CC BY\)](https://creativecommons.org/licenses/by/4.0/) license.

Keywords: flight data; LSTM; aerial systems; trajectory prediction; anomaly detection; risk assessment; resource-aware deployment; onboard deployment

1. Introduction

1.1. Background and Motivation

With the rapid development of aerospace technology and the unmanned aerial vehicle (UAV) industry, aircraft are increasingly being applied in complex tasks such as forest

health monitoring, maritime surveillance, agriculture, and search and rescue operations. However, aircraft operating in highly dynamic environments are exposed to various environmental disturbances, sensor failures, control system abnormalities, and mission-related uncertainties, which may trigger unexpected anomalies or accidents and consequently result in substantial losses. Therefore, the effective use of flight data for future-state prediction, abnormal-state identification, and operational risk assessment is of paramount importance for ensuring flight safety and evaluating the health status of flight systems. Flight data serve as the primary information carrier for recording the operational, kinematic, and attitude states of an aircraft and are typically collected by multiple airborne sensors, including Global Positioning System (GPS) receivers, gyroscopes, accelerometers, and other avionics systems. From a mathematical perspective, flight data are generally high-dimensional and non-stationary. Within sufficiently short and operationally homogeneous flight intervals, selected variables may be approximated as locally weakly stationary only when their estimated mean, variance, and autocovariance remain approximately invariant within the analysis window. During mission-phase transitions, high-maneuver operations, or sudden external disturbances, this local approximation may no longer hold, and the data may exhibit pronounced distribution shifts, nonlinear evolution, and complex cross-variable coupling.

Traditional methods for flight data analysis and safety-state assessment primarily rely on physics-based analytical models and shallow machine learning techniques. However, owing to the nonlinear and highly dynamic characteristics of UAV systems and the diversity of their operational missions, these methods often exhibit limited flexibility and insufficient adaptability when processing complex and high-dimensional flight data. They also face mathematical limitations in characterizing latent manifold structures [1]. In practical applications, UAV trajectories vary considerably across missions, operating environments, and maneuvering patterns. The resulting high-dimensional feature space may lead to the “curse of dimensionality”, reducing the effectiveness of conventional geometric distance metrics, such as Euclidean distance, in representing local similarities. Furthermore, variations in flight trajectories, environmental conditions, aircraft configurations, and mission phases can cause substantial distribution shifts in flight data, further degrading the generalization performance of models trained under fixed operating conditions.

To address these limitations, data-driven technologies represented by deep learning have attracted increasing attention because of their strong capability to approximate nonlinear dynamic sequences. For flight time-series data, Long Short-Term Memory (LSTM) networks and their variants have been widely employed because they can alleviate the long-term dependency and vanishing-gradient problems of conventional Recurrent Neural Networks (RNNs) and effectively capture the dynamic evolution of flight states. As research has progressed, the basic LSTM architecture has been extended to address multidimensional anomaly types—including point anomalies, bias faults, drift faults, stuck faults, and composite anomalies—as well as complex spatiotemporal correlations. For example, hybrid architectures integrating Convolutional Neural Networks (CNNs), such as CNN-LSTM and ConvLSTM, can jointly extract local spatial patterns and long-term temporal dependencies. LSTM-based autoencoders (LSTM-AEs), meanwhile, have demonstrated strong performance in the unsupervised reconstruction of multivariate time series and the detection of system-wide anomalies. Attention mechanisms, graph neural networks, physical constraints, uncertainty estimation, and lightweight model-compression techniques have also been incorporated into LSTM-family models to improve their accuracy, robustness, interpretability, and deployability.

Deep learning methods based on LSTM networks and their variant architectures have increasingly become mainstream approaches to flight data analysis and safety-state

assessment. Nevertheless, the existing literature still lacks a comprehensive review that simultaneously examines the architectural evolution, application paradigms, and engineering deployment of LSTM-family models driven by flight data. Previous reviews have predominantly focused on the broad categorization of anomaly detection methods or fault types, with limited discussion on the underlying architectural evolution associated with temporal dependency modeling, Euclidean manifold extraction, and non-Euclidean graph-space mapping. Moreover, trajectory prediction, anomaly detection, and risk assessment are often reviewed separately, despite their strong methodological and operational interdependence. A comprehensive evaluation framework is also required to jointly consider prediction and detection accuracy, robustness to distribution shifts, physical consistency, anomaly interpretability, uncertainty quantification, and computational resource constraints.

The three application domains were selected because they represent three complementary stages in the flight-safety analytics chain. Trajectory prediction estimates the future evolution of aircraft states; anomaly detection identifies deviations between observed or predicted behavior and normal operational patterns; and risk assessment further translates these deviations, together with their uncertainty and operational context, into interpretable risk levels and actionable warning lead times. These tasks may be implemented independently, but they can also form a progressive information flow from state forecasting to deviation identification and decision support. They share multivariate flight sequences and temporal representation models, while differing in target outputs, supervision requirements, evaluation metrics, and onboard deployment constraints. Moreover, the typical tasks introduced in Section 2.3 can be consolidated into these three higher-level categories, which jointly cover the predictive, diagnostic, and decision-support functions of flight-data-driven safety analysis.

Accordingly, this paper provides a comprehensive review and methodological evaluation of flight-data-driven LSTM-family models. Section 2 introduces the principal characteristics of flight data and the foundations of flight safety-state assessment. Section 3 analyzes the core mechanisms of LSTM from the perspective of temporal-state evolution and discusses its extension toward high-order spatiotemporal and topological architectures. Section 4 reviews the application paradigms of LSTM-family models in flight trajectory prediction, anomaly detection, and risk assessment, with particular emphasis on the connections and information flow among these three application levels. Section 5 develops a comprehensive evaluation framework for flight-data-driven models, covering data characteristics, methodological performance, physical-mechanism consistency, uncertainty and robustness, and resource-aware engineering constraints. Finally, the main conclusions, remaining challenges, and future research directions are presented in the concluding section.

1.2. Review Methodology

Literature sources and search strategy. The following digital libraries were searched: IEEE Xplore, Scopus, Web of Science, and ScienceDirect. The search period covered publications from 2018 to the first quarter of 2026, as the rapid development of LSTM-based flight-data modeling has occurred predominantly within this timeframe. Earlier foundational works were also considered where they provided essential background. The search strings combined three groups of keywords using Boolean operators. Google Scholar was used only for supplementary forward- and backward-citation tracking rather than as the principal bibliographic database.

The search strategy combined four concept groups: LSTM, aerial-system domain, flight-data application, and resource-aware deployment. A representative search expression was: (“long short-term memory” OR LSTM OR BiLSTM OR “bidirectional LSTM” OR “CNN-LSTM” OR ConvLSTM OR “LSTM autoencoder” OR “LSTM-AE” OR “attention

LSTM" OR "graph convolutional LSTM" OR "uncertainty-aware LSTM" OR "lightweight LSTM" OR "physics-informed LSTM") AND ("flight data" OR aircraft OR aviation OR UAV OR drone OR "unmanned aerial vehicle" OR QAR OR "quick access recorder" OR "ADS-B" OR "automatic dependent surveillance-broadcast") AND (trajectory OR "flight path" OR "state prediction" OR "flight delay" OR "fuel consumption" OR "wind estimation" OR clustering OR "anomaly detection" OR "fault detection" OR "fault diagnosis" OR "risk assessment" OR "risk prediction" OR "early warning" OR navigation OR "flight control").

A supplementary deployment-oriented search combined the first two concept groups with: ("onboard" OR "edge deployment" OR "real-time inference" OR "resource-constrained" OR SWaP OR "model compression" OR pruning OR quantization OR "knowledge distillation" OR FPGA OR accelerator OR "low-rank factorization").

Screening and selection process. A two-stage screening process was adopted. In the first stage, titles and abstracts were screened against the inclusion criteria. In the second stage, full texts were examined to confirm relevance and technical substance. The selection was iterative: keywords and inclusion criteria were refined as the scope of the review crystallized during the initial reading. Through this process, over 120 candidate records were initially identified from the database searches and snowballing. After screening, approximately 60 core studies were selected for detailed review and critical discussion. These studies were chosen to represent the major architectural categories and application paradigms in the field, rather than to constitute an exhaustive enumeration of all published works. A supplementary Web of Science search using the same keyword combinations was performed to cross-validate that the selected literature adequately covered the research landscape.

Quality assessment and synthesis approach. Because the reviewed studies exhibit substantial heterogeneity in data sources, aircraft platforms, flight phases, prediction horizons, anomaly definitions, data-partition strategies, baseline models, and evaluation metrics (as discussed in Section 6), a formal meta-analysis or quantitative performance ranking was not attempted. Instead, a narrative synthesis was adopted, organized around the hierarchical progression of LSTM architectures (Section 3) and the three interconnected application paradigms (Section 4). The methodological quality of each included study was assessed qualitatively based on: (i) the clarity and technical soundness of the problem formulation and model design; (ii) the adequacy of the dataset description, data preprocessing, and experimental protocol; (iii) the appropriateness of the baseline comparisons and evaluation metrics; and (iv) the specificity of the reported results, limitations, and deployment considerations. This qualitative assessment informed the critical commentary throughout the review, particularly in the comparative summaries (Section 3.5) and the multidimensional evaluation framework (Section 5).

Limitations of the review methodology. The selected studies primarily represent English-language publications from the major engineering and computer science databases. Publications in other languages or from regional journals may have been omitted. Additionally, the rapid growth of the field means that very recent work may not have been captured within the search period. These limitations are considered in the discussion on the review's scope in Section 6.

2. Flight Data Characteristics and Fundamentals of Safety-State Assessment

2.1. Sources, Types, and Key Parameters of Flight Data

Flight data serves as the core foundation for conducting trajectory prediction, state assessment, and safety analysis. Its completeness, accuracy, and multidimensionality directly determine the depth and reliability of safety analysis. The primary sources include the

airborne Quick Access Recorder (QAR), the Automatic Dependent Surveillance-Broadcast (ADS-B) system, specialized databases, and ground-based auxiliary information sources. Compared to traditional data, systems such as QAR and ADS-B are capable of providing high-precision three-dimensional or four-dimensional trajectory and state information at high frequencies [2].

Based on data types, flight data can be classified into five categories: spatial position data, motion state data, environmental data, system data, and operational data. Spatial position data includes longitude, latitude, altitude, and relative altitude; motion state data encompasses ground speed, airspeed, descent rate, and pitch angle; environmental data consists of wind speed, wind direction, air pressure, and temperature; system data involves engine speed and fuel flow; and operational data comprises flight plans, takeoff and landing information, and air traffic control commands. The key parameters encompass three-dimensional or four-dimensional coordinates, flight velocity, attitude angles, climb rate, track angle, convective weather features, and control command sequences to characterize the aircraft state, serving as the core input features for trajectory prediction, time-series analysis, anomaly detection, and safety assessment.

In addition to proprietary QAR records and operational ADS-B streams, several open research infrastructures have improved the reproducibility of flight-data modeling. The OpenSky Network provides large-scale crowdsourced air-traffic surveillance data, while TrajAir offers socially aware three-dimensional aircraft trajectories together with meteorological context in non-towered terminal airspace. More recently, TartanAviation has expanded benchmark coverage by providing synchronized image, air-traffic-control speech, and ADS-B trajectory data collected at both towered and non-towered airports [3–5].

2.2. Temporal Characteristics and Analysis Challenges of Flight Data

Fundamentally, flight data constitutes a high-dimensional, highly nonlinear, non-stationary, and multirate-coupled complex time series, characterized by strong temporal dependencies, drastic dynamic variations, and interwoven multidimensionality. For instance, the current position, velocity, and attitude evolve recursively from historical states, forming long-term temporal dependency chains. Furthermore, they are highly susceptible to environmental perturbations, exhibiting non-stationary and highly stochastic fluctuations. Furthermore, the sampling rates of different airborne recording systems are often inconsistent. For instance, the sampling rates for climb rate and throttle data may be 2 Hz, while those for roll and pitch angles can reach up to 25 Hz [6]. This easily results in a mixture of multirate temporal sequences, constituting a typical Multirate Multivariate Time Series. Naturally, when subjected to faults, interference, or occlusion, the data frequently exhibits anomalies, abrupt jumps, and missing values.

The difficulties in its time-series analysis are primarily manifested in three aspects. First, the difficulty in modeling long-term temporal dependencies; traditional time-series methods, such as Markov models, struggle to capture kinematic features and evolutionary patterns over long-distance flights. Second, the complexity of nonlinear coupling relationships; the intricate interactions among position, velocity, attitude, and environmental factors make it difficult to establish precise analytical models. Third, the intensity of dynamic perturbations and the high risk of overfitting; uncontrollable factors render the data distribution susceptible to temporal shifts. Furthermore, the massive, high-dimensional data inherently contain numerous non-stationary signals and redundant parameters; without effective feature selection and decomposition, the generalization capability of the models is severely constrained.

2.3. Typical Tasks in Flight Safety-State Assessment

Flight safety-state assessment constitutes a comprehensive framework grounded in flight data that integrates time-series analysis, feature extraction, pattern recognition, and risk modeling to execute dynamic quantitative evaluations, anomaly identification, risk anticipation, and classification regarding operational safety, stability, and reliability. Its typical tasks encompass five primary directions: trajectory compliance evaluation, anomalous maneuver identification, temporal anomaly detection, dynamic risk prediction, and safety level classification.

Flight safety-state assessment encompasses trajectory compliance and future-state estimation, anomalous maneuver and temporal anomaly identification, and dynamic risk prediction and safety level classification. In this review, these tasks are consolidated into three higher-level application categories: trajectory prediction, anomaly detection, and risk assessment. They correspond to forecasting, diagnosis, and decision support, respectively, and may form a progressive chain from future-state estimation to deviation identification and risk-oriented decision-making [2,6–10].

3. Evolution of LSTM-Family Architectures for Flight-Data Modeling

This section reviews the evolution of LSTM-family architectures through a hierarchical progression from temporal sequence modeling to increasingly comprehensive spatial and system-level representations. Section 3.1 establishes the fundamental temporal representation capability by examining how LSTM and BiLSTM capture long-term dependencies and bidirectional contextual information in flight sequences. However, purely recurrent architectures remain limited in representing local geometric patterns and short-term spatiotemporal variations. To address this limitation, Section 3.2 introduces convolutional feature extraction to extend temporal modeling toward local Euclidean spatiotemporal representation. Nevertheless, Euclidean convolution alone cannot adequately characterize the irregular connectivity and interaction patterns among aircraft, airports, routes, and airspace sectors. Section 3.3 therefore further expands the modeling scope to non-Euclidean graph spaces, enabling system-level representation of aviation network topology and inter-node dependencies. As these increasingly complex architectures introduce longer-range dependencies and higher computational costs, Section 3.4 integrates attention-based global dependency modeling with computational dimensionality-reduction strategies to improve feature focusing and support resource-constrained deployment. Overall, these subsections form a progressive pathway from temporal dependency modeling, through local spatiotemporal feature extraction and system-level topological representation, to global information integration and computational efficiency, thereby providing the architectural foundation for the application review in Section 4 and the multidimensional evaluation framework in Section 5. Overall, these subsections form a progressive pathway from temporal dependency modeling, through local spatiotemporal feature extraction and system-level topological representation, to global information integration and computational efficiency, as summarized in Figure 1.

3.1. Temporal Dependency Modeling

LSTM provides the fundamental temporal representation capability for flight-data modeling by selectively preserving historical information over long sequences. Compared with conventional RNNs and short-memory statistical models, it is better suited to describing the continuous evolution of aircraft position, velocity, attitude, and system states. Its relatively compact recurrent structure also makes it a practical baseline for trajectory prediction, state estimation, and onboard monitoring. However, standard LSTM relies only on

past observations and has limited ability to represent local spatial patterns, cross-variable interactions, and abrupt maneuvering dynamics.

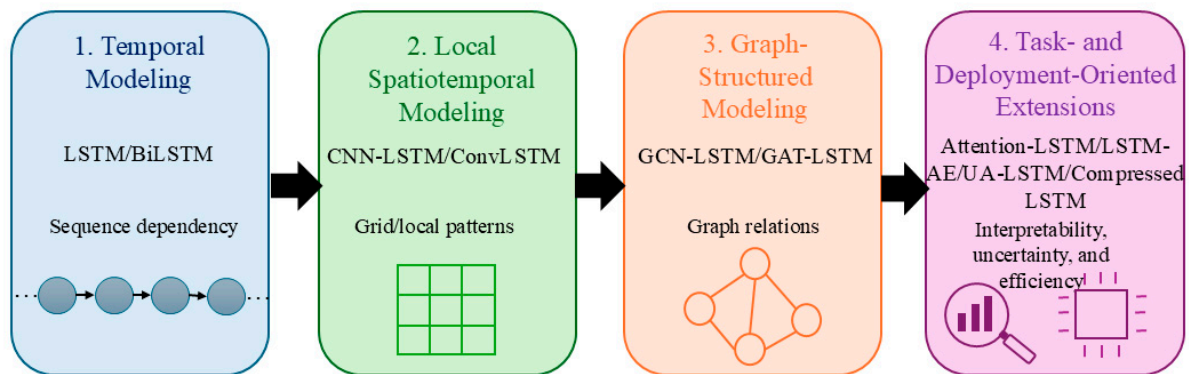


Figure 1. Evolution of the LSTM Family.

BiLSTM extends this temporal representation by processing flight sequences in both forward and backward directions. The additional reverse context is beneficial for off-line flight-phase recognition, post-flight trajectory reconstruction, and anomaly diagnosis, where complete flight sequences are available. Nevertheless, BiLSTM requires access to a complete input window and therefore cannot support strictly streaming sample-by-sample inference without buffering. When the backward branch incorporates observations occurring after the target time, it may also violate causal inference requirements. Nevertheless, it can still be applied to online forecasting based on completed historical windows, albeit with increased latency and memory consumption. It also increases the parameter count, memory access, and inference cost compared with a unidirectional LSTM.

Therefore, the choice between LSTM and BiLSTM depends primarily on the availability of future context and the deployment mode. Standard LSTM is generally more suitable for causal onboard prediction and low-latency monitoring, whereas BiLSTM is preferable for offline analysis tasks that prioritize complete contextual representation over real-time inference efficiency.

In practical engineering applications for UAV path prediction, raw sensor data is frequently constrained by non-uniform sampling rates, noise interference, and nonlinear perturbations. To address this pain point, an EC-Bessel-BiLSTM real-time prediction model integrating PID error compensation was proposed in Ref. [11]. This method employs Bessel curve fitting for the smooth correction of raw GPS data, effectively eliminating dimensional inconsistencies and high-frequency noise interference. Building upon this, the bidirectional LSTM architecture is utilized to capture the spatiotemporal dependency features of the trajectory, and a PID feedback compensation module is innovatively introduced to dynamically correct the prediction results. This significantly mitigates the lag inherent in deep learning models during sudden maneuvering scenarios, achieving a sub-meter prediction accuracy while maintaining an ultra-fast response time of under 0.1 seconds, thereby providing a highly real-time and robust integrated technical solution for autonomous UAV operation monitoring. Because the backward branch requires a completed input window, this implementation should be interpreted as window-buffered near-real-time prediction rather than strictly causal sample-by-sample inference. Its suitability for onboard control therefore depends on the input-window length, buffering delay, and available computing resources.

3.2. Local Spatiotemporal Representation

In flight trajectory prediction tasks, trajectory data fundamentally manifests as a complex, non-stationary temporal manifold embedded within high-dimensional Euclidean space. This manifold deeply couples multidimensional continuous kinematic variables—such as longitude, latitude, altitude, velocity, heading, and pitch—and is characterized by strong nonlinearity, dynamic spatiotemporal coupling, and sensitivity to local geometric trends. Traditional time-series models, such as LSTM, while adept at capturing temporal dependencies, exhibit limited capabilities in characterizing spatial morphology, local geometric features, and kinematic patterns, making it difficult to fully mine the spatial structures embedded within trajectories. Therefore, introducing convolutional feature extraction mechanisms to model the geometric manifold of trajectories from a Euclidean space perspective can effectively compensate for the inadequacies of time-series models in spatial representation. This enables the joint modeling of temporal dependencies and spatial morphology, laying a solid foundation for subsequent deep feature learning and high-order topological construction.

3.2.1. 1D Convolution and Trajectory Manifold Mapping

Unlike the 2D convolutions used for images where spatial translation invariance is key, flight trajectories typically manifest as multivariate vector sequences that evolve over time, such as longitude, latitude, altitude, velocity, and heading.

For this type of time-series data, a one-dimensional convolutional neural network (1D-CNN) can be employed to perform localized weighted scanning within a temporal window, achieving feature compression of the trajectory's local trends and geometric morphology [12]. Through weight sharing and feature fusion by convolutional kernels within local temporal windows, this mechanism maps the raw trajectory data into a high-dimensional abstract feature space, thereby capturing the local trend and geometric morphology features of the trajectory.

To address highly dynamic maneuvering scenarios in air combat, a CNN-LSTM model integrating an attention mechanism and a social pooling module was proposed in Ref. [13]. This study employs a 1D-CNN to capture the local spatial geometric features of fighter trajectories, integrates a social pooling module to aggregate the multi-aircraft interaction space, and utilizes an attention mechanism to focus on mutated trajectory features in real time. Experimental validation demonstrates that when handling complex maneuvers, the proposed method achieves improvements of 32% and 34% in Average Displacement Error (ADE) and Final Displacement Error (FDE), respectively, compared to traditional models, fulfilling the requirements for real-time decision-making while ensuring high accuracy.

3.2.2. Nested Architecture and Deep Feature Embedding

Convolution-enhanced LSTM architectures can generally be divided into CNN-LSTM and ConvLSTM [14] according to the integration depth between convolutional and recurrent operations. In CNN-LSTM, convolutional layers serve as independent front-end modules that extract local variation patterns from multivariate flight sequences, after which LSTM models their long-term temporal evolution. This modular architecture is relatively easy to construct and train and is suitable for trajectory segments, sensor windows, and other one-dimensional multivariate sequences. By contrast, ConvLSTM incorporates convolutional operations directly into recurrent state updating, allowing local spatial structures to be retained throughout temporal propagation. It is therefore more suitable for regularly structured spatiotemporal inputs, such as meteorological grids and weather-feature fields, but generally requires greater memory access and computational resources.

In aerial-system applications, CNN-LSTM models have been employed for aircraft four-dimensional trajectory prediction by combining local kinematic feature extraction with long-term temporal modeling [15]. Convolution-embedded LSTM architectures have also been applied to trajectory prediction under complex meteorological disturbances [16]. In these models, convolutional layers extract local weather and trajectory characteristics, while recurrent or sequence-to-sequence structures model their temporal evolution [17]. These studies indicate that convolutional–recurrent fusion can improve the representation of short-term trajectory variations, maneuvering patterns, and weather–trajectory interactions that cannot be adequately captured by conventional LSTM alone.

The choice between CNN-LSTM and ConvLSTM should depend on the organization of the input data and the deployment requirements. CNN-LSTM is generally preferable when the inputs are multivariate temporal sequences and implementation modularity, training simplicity, and computational controllability are prioritized. ConvLSTM is more appropriate when the data retain explicit grid-like spatial relationships and the preservation of spatial information during recurrent propagation is essential. However, its potential performance improvement should be evaluated together with parameter count, memory footprint, inference latency, and onboard computing constraints. For both architectures, robustness under non-stationary flight conditions further depends on appropriate feature preprocessing, normalization, and representation-stabilization strategies, as discussed in Section 3.2.3.

3.2.3. Distinguishing Statistical Stationarity, Optimization Stability, Smoothness, and Normalization

Statistical stationarity, numerical or optimization stability, temporal smoothness, and feature normalization describe different properties of flight-data modeling and should not be used interchangeably. Statistical stationarity is a property of the underlying stochastic process. Under the assumption of finite second-order moments, a process X_t is weakly stationary only when its mean and variance are time-invariant and its autocovariance depends on the time lag rather than on the absolute time; that is,

$$E[X_t] = \mu, \quad (1)$$

$$\text{Var}(X_t) = \sigma^2, \quad (2)$$

and

$$\text{Cov}(X_t, X_{t+\tau}) = \gamma(\tau). \quad (3)$$

Because flight states vary across mission phases and environmental conditions, weak stationarity can generally be regarded only as a local approximation within sufficiently short and operationally homogeneous time windows.

Numerical or optimization stability instead describes the sensitivity of model training and inference to input perturbations, parameter initialization, gradient propagation, and finite-precision computation. Nonlinear activation functions such as ReLU or GeLU and normalization operations such as batch normalization or layer normalization may improve gradient propagation and training consistency. However, these operations do not make the underlying flight-data process statistically stationary. Feature normalization only recenters or rescales the input variables or intermediate representations, whereas temporal smoothness describes the magnitude of local variation between adjacent observations. Consequently, a normalized or smoothed sequence is not necessarily stationary.

Signal-decomposition methods may simplify the representation of non-stationary flight data without providing a stationarity guarantee. For example, the CEEMDAN-based preprocessing adopted in [8] decomposes the original signal into intrinsic mode functions

associated with different temporal scales. This decomposition may reduce mode mixing and facilitate subsequent feature learning, but the resulting components should not be described as stationary unless their mean, variance, and autocovariance properties are explicitly verified. Similarly, the TCN-KAN framework in [18] can capture multiscale local temporal patterns, but it does not inherently guarantee stationary feature distributions.

Therefore, recurrent gating, convolutional feature extraction, activation functions, normalization, and multiscale decomposition should be regarded as complementary representation-stabilization strategies rather than stationarity-guarantee mechanisms. The studies in [13–16] reported improved representation or prediction performance under their respective experimental settings, but these empirical observations do not establish general mathematical guarantees of stationarity or numerical stability. Claims regarding statistical stationarity should be supported by explicit assumptions and statistical verification, whereas claims regarding numerical stability should be supported by perturbation, convergence, gradient, or repeated-training analyses.

3.3. Topological Extension in Non-Euclidean Graph Spaces

Within an air traffic system, aircraft, route nodes, and airspace sectors do not exist in isolation; rather, they constitute a dynamic network characterized by a complex topological structure. Traditional Euclidean space modeling methods are often inadequate for capturing route connectivity, sector adjacency relationships, and the spatial propagation characteristics of traffic flow. Through topological extension in non-Euclidean graph spaces, integrating the discrete topological features of the aviation system into deep learning frameworks enables the precise modeling of spatiotemporal correlations within aviation networks.

Graph-based LSTM architectures are particularly valuable when flight states are influenced by explicit relational structures, such as route connectivity, sector adjacency, airport interactions, or multi-aircraft coordination. Their performance, however, depends heavily on graph construction. Geographical adjacency, operational connectivity, statistical correlation, and learned dynamic adjacency may produce substantially different representations. Static graphs are computationally efficient but may fail to reflect rapidly changing traffic interactions, whereas dynamic graphs offer greater adaptability at the cost of additional computation and reduced interpretability. Future studies should therefore report graph-definition criteria explicitly and evaluate the transferability of graph structures across airports, airspace regions, and operating conditions. Recent aviation-network studies further demonstrate that graph construction should reflect both physical and operational relationships. GOGCN jointly represents geographical proximity and operational interactions among airports, whereas AG2S-Net combines graph convolution, BiLSTM, sequence-to-sequence learning, and attention for network-wide, multi-step, flight-delay prediction. More recent multi-attention graph models further characterize long-horizon and dynamically varying airport dependencies [19–21]. Because these models were evaluated using different airport networks, prediction horizons, and data partitions, they are used here to illustrate alternative graph-design strategies rather than to establish a unified performance ranking.

3.3.1. Graph Construction for Aviation Networks

The air traffic network can be abstracted as a weighted directed graph $G = (V, E)$ composed of nodes and edges, where the nodes V represent airports, route intersections, or control sectors, and the edges E characterize the physical connections or logical associations between them. In sector traffic flow forecasting, this mapping encompasses not only geographical adjacency relationships but also directional flow dependencies based on

route planning. For example, in their study of airspace sector traffic flow, the topological structures of the sector's interior and its associated route nodes were mapped into a graph, thereby overcoming the limitations of previous approaches that focused exclusively on a sector's internal traffic volume while neglecting interactions with neighboring sectors Ref. [22]. This mapping approach transforms the non-stationary physical connections of the aviation network into a graph-based adjacency matrix, providing a foundational topology for the subsequent mining of system-level spatial dependencies.

By utilizing Graph Convolutional Networks (GCNs) [23] to process adjacency matrices within non-Euclidean spaces, the model captures the dynamic spatiotemporal interactions between sectors. Through the GCN layer, the model can learn the topological features of the sector's spatial structure, utilizing not only the temporal information of a single sector but also explicitly correlating the traffic flow load conditions of adjacent sectors. Furthermore, the topology of the traffic system is not static; driven by traffic flow shifts and meteorological impacts, the mutual influence between nodes fluctuates over time. By introducing dynamic graph convolutions into the temporal network, the model can more precisely capture the variations in topological connection strength induced by unexpected events. Beyond conventional graph convolution, graph attention networks assign adaptive weights to neighboring nodes and provide a flexible mechanism for representing heterogeneous interactions among aircraft, airports, routes, and sectors [24].

3.3.2. Graph–Temporal Fusion Architectures

Traditional time-series models are prone to losing hierarchical relationships between nodes when processing non-Euclidean spatial data. Conversely, GCNs can directly extract complex dependencies within non-Euclidean spaces. By performing spectral convolution or spatial domain aggregation on the graph structure, the model can capture both node features and their corresponding neighborhood information. First, the GCN completes the aggregation of spatial dependencies within the graph domain, generating spatial vectors with topological semantics; subsequently, the LSTM utilizes these topological embeddings as an extension of the temporal input, thereby preserving global spatial consistency within the state transition matrix.

Taking the AGC-LSTM [22] architecture as an example, the model processes spatial topological relationships by incorporating a GCN layer, and integrates an attention mechanism to dynamically allocate traffic impact weights across different sectors. Experiments demonstrate that this nested architecture effectively extracts long-range spatial interaction features between sectors. Compared to purely temporal models, it exhibits significant predictive robustness in scenarios involving complex and abrupt changes in sector traffic flow. Furthermore, in airport system delay prediction, multi-attribute airport node information was aggregated through graph neural networks to effectively model the propagation mechanism of delays across the airport network Ref. [25]. This demonstrates the core role of graph deep learning in extracting correlated features within complex systems. At the network-wide level, spatiotemporal propagation learning has also been used to characterize how local delays spread across interconnected airports and routes, further demonstrating the importance of jointly modeling temporal evolution and network structure [26].

3.3.3. Stability and Regularization in Dynamic Graph Modeling

Dynamic changes in traffic flow, route availability, and meteorological conditions may cause graph representations to vary substantially over time. Adaptive adjacency learning, graph–temporal fusion, and graph regularization can reduce sensitivity to short-term perturbations and improve training consistency, but they do not guarantee either statistical stationarity of the input process or numerical stability of the complete network.

Within dynamic aviation networks, the evolution of topological structures and the non-stationary characteristics of traffic flow frequently lead to oscillations during model training. To improve training and representation stability, existing graph-based architectures commonly introduce the following regularization strategies:

- (1) **Adaptive Adjacency Learning:** Attention-based graph operators dynamically update inter-node weights according to the current traffic state. This mechanism can improve the adaptability of graph representations to time-varying interactions, but its stability depends on the learned weight distribution, training procedure, and magnitude of input perturbations. Graph WaveNet demonstrates that adaptive adjacency learning can represent time-varying spatial dependencies without relying exclusively on a fixed predefined graph [27].
- (2) **Multiscale Graph–Temporal Fusion:** Cascading graph convolution with LSTM enables spatial relationships and temporal evolution to be modeled jointly. Recurrent memory may attenuate the influence of short-lived variations in graph features, but it does not ensure that the hidden-state sequence is statistically stationary, numerically stable, or uniformly bounded.
- (3) **Graph Regularization:** Laplacian regularization penalizes large differences between the representations of connected nodes and therefore encourages local feature consistency. It should not be interpreted as enforcing statistical stationarity or guaranteeing network convergence. Such claims would require explicit assumptions and corresponding perturbation or convergence analyses.

3.4. Global, Probabilistic, and Resource-Aware Extensions

Section 3.4 examines extensions that address global dependency modeling, reconstruction-based anomaly detection, uncertainty representation, and resource-constrained deployment. Attention mechanisms assign adaptive weights to critical time steps or attributes, reconstruction-aware models identify anomalies through residuals, and uncertainty-aware models estimate predictive intervals or distributions. Informer and Autoformer are retained only as non-LSTM comparison baselines for long-sequence forecasting [28,29].

3.4.1. Attention-Enhanced LSTM

A single recurrent gating structure struggles to automatically identify temporal importance within long sequences. By introducing an attention mechanism, the model can achieve a paradigm shift from local recursive processing to global feature capture.

Standard LSTM lacks an explicit mechanism for assigning global importance weights directly to all time steps or input attributes. Attention mechanisms address this limitation by adaptively emphasizing temporally or physically relevant information. However, attention weights should not automatically be interpreted as causal explanations, particularly when flight variables are strongly correlated. To address the characteristic that different attributes within flight track data—such as altitude, airspeed, pitch angle, and thrust—exert varying degrees of impact on risk, this research introduces an attribute control gate. By calculating the interactive relationships among attributes within a time window, the model can dynamically adjust the contribution weights of different parameters, thereby achieving a “weighted focus” on critical attributes during the feature extraction phase. This achieves the leap from local state recurrence to the capture of global critical states [30].

3.4.2. Reconstruction-Aware LSTM

Reconstruction-aware LSTM architectures, represented by LSTM-AE and VAE-LSTM [31], learn the temporal structure of normal flight sequences and identify anomalies according to reconstruction residuals. Their main advantage is their reduced dependence

on labeled fault samples, which are often scarce in aviation systems. However, their performance depends strongly on reconstruction-threshold selection, training-data purity, and the ability of the latent space to prevent abnormal samples from being reconstructed as normal patterns.

3.4.3. Uncertainty-Aware LSTM

Uncertainty-aware LSTM models extend deterministic prediction by estimating confidence intervals or predictive distributions. This capability is particularly important for risk assessment because identical prediction errors may have different safety implications under different flight conditions. Nevertheless, uncertainty estimates must be calibrated across flight phases and operating environments, and sampling-based approaches may introduce additional inference costs.

3.4.4. Feature Reduction, Model Compression, and Hardware-Aware Deployment

Resource-aware deployment should be considered at three interconnected levels: input-space reduction, model-level compression, and hardware-aware acceleration. These levels address different sources of computational overhead and should not be treated as equivalent. Input-space reduction decreases the number of flight variables processed by the network, model compression directly reduces the storage and computation required by the recurrent architecture, and hardware acceleration improves the execution efficiency of the compressed model on embedded platforms.

At the input level, feature-selection and projection methods can remove redundant or weakly informative variables before recurrent modeling. Representative strategies include maximum relevance and minimum redundancy selection, kernel principal component analysis, physics-informed correlation filtering, and singular value decomposition. For example, KPCA- and mRMR-based preprocessing have been used to reduce high-dimensional QAR variables while retaining information related to fuel consumption and landing risks [8,10]. Such methods can decrease input dimensionality, suppress noise, and reduce memory traffic. However, input reduction alone does not necessarily decrease the parameter count or computational complexity of the recurrent layers. Excessive feature elimination may also remove weak but safety-relevant fault signatures.

At the model level, pruning, low-rank factorization, quantization, and knowledge distillation provide complementary approaches for constructing lightweight LSTM-family architectures. Weight or kernel pruning removes redundant parameters and recurrent connections, whereas low-rank factorization decomposes large state-transition matrices into smaller matrices with reduced computational cost. Quantization decreases the numerical precision of weights and activations, and knowledge distillation transfers the representation capability of a large teacher model to a compact student model. These methods can reduce model size, floating-point operations, memory access, and inference latency. Nevertheless, aggressive compression may degrade trajectory-prediction accuracy, reduce sensitivity to small-amplitude faults, or affect the calibration of probabilistic risk estimates. Compression performance should therefore be evaluated jointly using the compression ratio, parameter count, memory footprint, inference latency, energy consumption, and task-specific accuracy metrics. Established compression methods provide methodological foundations for recurrent edge deployment. Deep Compression combines network pruning and trained quantization to reduce parameter storage and memory access, while integer-only quantization enables efficient inference on fixed-point hardware. FastGRNN further demonstrates that low-rank, sparse, and quantized recurrent matrices can be integrated directly into the recurrent architecture to reduce model size and inference cost [32–34]. However, compres-

sion results obtained from non-aviation benchmarks should be interpreted as transferable methodological evidence rather than as direct proof of flight-data performance.

At the hardware level, field-programmable gate arrays, digital signal processors, and embedded graphics processing units can exploit temporal and matrix-level parallelism to accelerate recurrent inference. However, hardware acceleration cannot fully compensate for an excessively large or structurally redundant model. Algorithm–hardware co-design is therefore required to align the network architecture, numerical precision, memory-access pattern, and parallel execution strategy with the physical limitations of the target on-board platform. For aerial edge systems, the evaluation should report not only prediction or detection performance, but also model parameters, memory consumption, FLOPs or MACs, end-to-end latency, throughput, power consumption, and size, weight, and power constraints, as further discussed in Sections 5.3 and 5.4. Representative hardware implementations further illustrate these acceleration strategies. ESE demonstrates the joint use of pruning, quantization, sparse scheduling, and FPGA acceleration for LSTM inference, whereas EdgeDRNN exploits temporal sparsity to support low-latency recurrent inference on an edge-oriented FPGA platform [35,36]. These studies provide direct evidence that recurrent-network acceleration can reduce memory traffic and inference latency. Nevertheless, their results should not be interpreted as onboard aviation evidence unless the complete model is evaluated on the intended aircraft platform under representative input rates, numerical precision, and power constraints.

Overall, feature reduction, model compression, and hardware acceleration constitute complementary rather than interchangeable optimization stages. A compact causal LSTM may be preferable for low-latency trajectory prediction, whereas continuous anomaly monitoring requires compression strategies that preserve sensitivity to weak faults and abrupt state changes. For uncertainty-aware risk assessment, lightweight deployment should additionally maintain probabilistic calibration and reliable confidence boundaries. Therefore, the objective of resource-aware design is not simply to minimize model size, but to achieve an application-specific balance among representation capability, safety performance, computational cost, and onboard deployment feasibility.

3.5. Comparative Summary and Model-Selection Guidance

The evolution of LSTM-family models does not imply that increasingly complex architectures are universally superior. Standard LSTM remains suitable for causal temporal modeling under limited computing resources, whereas BiLSTM is more appropriate for offline analysis requiring complete sequence context. CNN-LSTM and ConvLSTM improve local spatiotemporal representation but introduce additional computational overhead. Graph-based LSTM models are advantageous for system-level aviation networks with explicit relational structures, although their effectiveness depends strongly on graph construction and transferability. Attention-enhanced models improve the identification of critical time steps and attributes, whereas uncertainty-aware architectures are more suitable for probabilistic risk assessment. Lightweight and compressed LSTM architectures can reduce memory occupation and inference latency for onboard implementation, but their compression ratios should be balanced against accuracy degradation, fault sensitivity, and uncertainty calibration. For edge deployment, architectural selection should therefore jointly consider representation capability, task requirements, data structure, causality, interpretability, compression tolerance, and hardware constraints rather than relying solely on predictive accuracy.

Figure 2 summarizes the correspondence among flight-data structures, representative LSTM-family architectures, application tasks, and deployment constraints. Sequential flight states are primarily associated with LSTM or BiLSTM models, whereas local tempo-

ral patterns, relational networks, reconstruction residuals, and long-range dependencies motivate convolution-enhanced, graph-based, autoencoder-based, and attention-enhanced architectures, respectively. The mapping further indicates that architecture selection should be driven by the organization of the input data and the operational requirements rather than by model complexity alone.

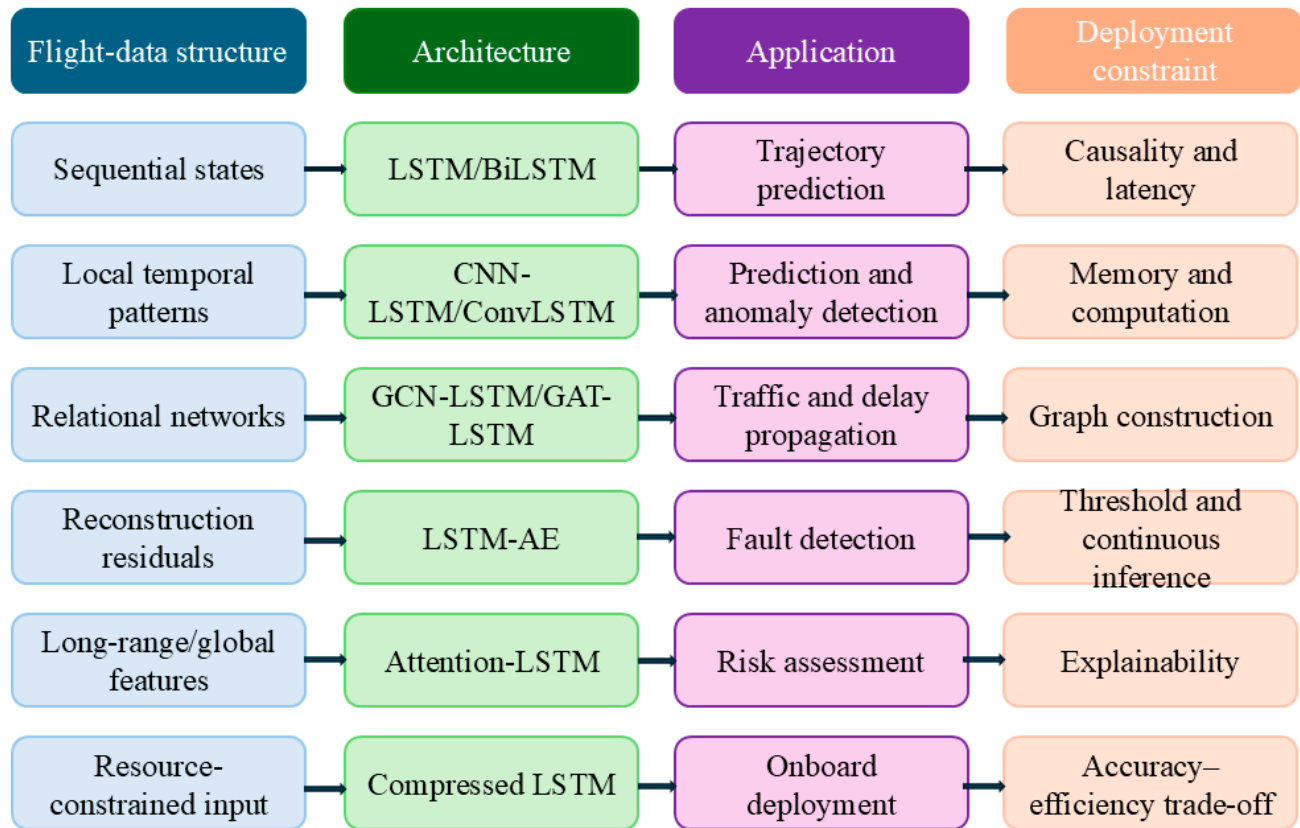


Figure 2. Mapping between flight-data structures, LSTM-family architectures, application tasks, and deployment constraints.

To complement the cross-layer mapping shown in Figure 2, Table 1 provides a more detailed comparison of the main capabilities, suitable applications, advantages, limitations, and deployment considerations of representative LSTM-family architectures.

Table 1. Comparative characteristics of LSTM-family architectures for flight-data modeling.

Architecture	Main Capability	Suitable Applications	Main Advantage	Key Limitation	Deployment Consideration
LSTM	Causal temporal dependency	Real-time trajectory prediction, monitoring	Compact and causal	Weak spatial representation	Relatively low cost
BiLSTM	Bidirectional context	Offline diagnosis, phase recognition	Complete sequence context	Requires a complete input window and may violate strict causality	Higher latency and memory
CNN-LSTM	Local feature extraction plus temporal modeling	Trajectory and anomaly prediction	Modular and easy to implement	Kernel and window sensitivity	Moderate cost

Table 1. Cont.

Architecture	Main Capability	Suitable Applications	Main Advantage	Key Limitation	Deployment Consideration
ConvLSTM	Joint recurrent spatiotemporal modeling	Weather-coupled prediction	Preserves local spatial structure	High computation	Moderate-to-high cost
LSTM-AE	Unsupervised reconstruction	Anomaly detection	Reduces label dependence	Threshold sensitivity	Continuous inference required
GCN-LSTM	Non-Euclidean relational modeling	Airport/sector traffic prediction	Captures network propagation	Graph construction dependence	High memory and computation
Attention-LSTM	Global temporal/attribute focusing	Risk warning and long sequences	Highlights critical features	Weight does not always equal explanation	Additional computation
Uncertainty-aware LSTM	Probabilistic output	Risk assessment	Provides confidence bounds	Calibration difficulty	Repeated sampling may be costly
Lightweight/Compressed LSTM	Lightweight inference	Onboard deployment	Reduces memory and latency	Possible accuracy loss	Hardware-dependent

The deployment-related indicators explicitly reported in the reviewed flight-data studies are summarized in Section 5.3 and Table 5. Because the available evidence varies substantially across publications, this review distinguishes four levels of validation: offline algorithmic evaluation, offboard real-time execution, simulation-based closed-loop validation, and direct onboard implementation. Claims of real-time, lightweight, online, or edge-deployable performance are not treated as evidence of onboard feasibility unless they are supported by quantitative measurements on a clearly identified target platform.

4. Evolution and Application Paradigms of LSTM Variants Driven by Flight Applications

This section organizes flight-data-driven applications into three interconnected categories. Trajectory prediction provides estimates of future aircraft states and expected operational evolution. Anomaly detection compares actual or predicted states with normal patterns to identify deviations and possible faults. Risk assessment combines predicted trajectories, detected anomalies, uncertainty information, and operational constraints to determine event severity and warning lead time. Therefore, the three categories correspond to forecasting, diagnosis, and decision support, respectively, while sharing a common flight-data and temporal-modeling foundation.

4.1. Trajectory and State Prediction

Flight-data-driven time-series prediction is a core technology for aviation safety and intelligent air traffic management. Owing to their strong capability in capturing long-term temporal dependencies, LSTM and its variants have become mainstream methods for tasks such as flight delay prediction, UAV trajectory prediction, and civil aircraft trajectory prediction. Research based on LSTM and its variants has evolved from basic time-series modeling into diverse research paradigms involving spatiotemporal multidimensional fusion, online dynamic updating, coupling with physical mechanisms, and efficient model pruning. Public benchmarks are also shifting aircraft-trajectory prediction from isolated proprietary datasets toward more reproducible evaluation. TrajAir provides socially interactive three-dimensional trajectories for non-towered terminal airspace, enabling the evaluation of models under aircraft–aircraft and aircraft–environment interactions. Recent probabilistic approaches based on conditional flow matching further generate multiple plausible ADS-B

trajectory futures rather than a single deterministic path [4,37]. Although such generative models do not belong to the LSTM family, they provide useful comparative baselines for evaluating uncertainty representation, trajectory diversity, and computational efficiency.

4.1.1. Spatiotemporal Feature Fusion Mechanism: From Local Recurrence to Global Modeling

To overcome the prediction limitations of traditional models that rely solely on temporal features, existing studies have introduced spatiotemporal attention mechanisms to enhance the network's perception of external variables such as meteorological conditions and route topology. An Att-Conv-LSTM model was proposed in Ref. [38], employing a hybrid convolutional LSTM architecture to integrate spatial features extracted from meteorological data with the temporal sequence features of flight delays. By introducing a temporal attention module, the model automatically assigns weights to critical delay-related time steps. In an empirical analysis of four airports in the Beijing–Tianjin–Hebei region, the RMSE was successfully reduced from 17.52 min to 10.31 min. Within the experimental setting of Ref. [38], the results support the value of jointly modeling meteorological and temporal features for flight-delay prediction.

4.1.2. Embedding of Physical Constraints

Purely data-driven models often face challenges such as divergent trajectory predictions or violations of aerodynamically reasonable flight dynamics. A constrained LSTM network incorporating flight-physics constraints was proposed in Ref. [39], in which the dynamic characteristics of different civil aviation flight phases, including climb, cruise, and approach, are transformed into regularization terms in the loss function to ensure the physical validity of the predicted trajectories. This method uses the DBSCAN [40] algorithm to automatically identify flight phases and dynamically switch the corresponding constraint terms. Empirical results show that the proposed network reduces the average altitude error over the entire flight to 9.96 m, representing an 82% reduction compared with the standard LSTM. Within this study, physical regularization reduced trajectory divergence and improved consistency with phase-specific flight dynamics.

4.1.3. Adaptive Online Incremental Learning and Abrupt Trajectory Change Tracking

For high-dynamic environments, such as the degradation of model generalization caused by UAV maneuvering flight, researchers have proposed dynamic updating strategies based on online incremental learning. Within the experimental setting of Ref. [41], the incremental RLSTM achieved a three-dimensional prediction error of 6.25 m and outperformed the LSTM and BiLSTM baselines evaluated using the same data and configuration. This result supports the usefulness of online updating under the maneuvering conditions examined in that study, but it does not establish cross-study superiority.

4.1.4. Efficiency Optimization Paradigm: Trade-Off Between Computational Complexity and Prediction Accuracy

In resource-constrained or small-sample environments, highly parameterized LSTM models suffer from high computational overhead and the risk of overfitting, giving rise to differentiated model-selection strategies. Strongly correlated feature variables were selected through correlation-based feature analysis, and Sta_POS NARX, a nonlinear autoregressive exogenous model, was proposed as an alternative solution in Ref. [42]. The study found that attitude angles exhibit significantly stronger correlations with trajectory positions than traditional control signals. Using the Sta_POS NARX model, the training efficiency was improved by three to five times while maintaining a prediction R^2 of 0.99. This finding indicates that deeper model architectures are not necessarily superior; instead, targeted

optimization of input features and computational models constitutes a core engineering principle for real-time aviation prediction tasks.

4.2. Abnormal State Detection

UAV flight data serve as a core carrier for characterizing system states. Anomaly detection techniques can identify sensor faults, deviations in command execution, control abnormalities, and environmental disturbances in real time, thereby playing a critical role in ensuring flight safety and reducing accident risks. With the widespread application of UAV technologies in agriculture, power-line inspection, disaster rescue, and other fields, operational safety issues have become increasingly prominent.

Traditional anomaly detection methods mainly include knowledge-driven and model-driven approaches. Knowledge-driven methods rely on expert experience to formulate rules and thus exhibit poor generalization capability, whereas model-driven methods require accurate physical models and are difficult to adapt to complex UAV systems. In recent years, data-driven methods have become the mainstream research direction in this field, owing to their advantages of requiring no prior knowledge and automatically extracting complex features.

4.2.1. Unsupervised Anomaly Detection Based on LSTM-AE

LSTM networks are well suited for processing time-series data, while AEs identify anomalies based on reconstruction errors. The combination of these two models, namely LSTM-AE, has become one of the most widely used unsupervised anomaly detection frameworks. The LSTM-AE framework captures and reconstructs the temporal features of normal flight patterns, and then performs anomaly discrimination according to the magnitude of the reconstruction error.

Methodological developments in adjacent aerospace and multivariate time-series domains have also influenced reconstruction-based flight anomaly detection. Representative studies include LSTM-based spacecraft anomaly detection with nonparametric dynamic thresholding [43], deep autoencoding Gaussian mixture models that jointly learn latent representations and density structures [44], and stochastic recurrent models for robust multivariate time-series anomaly detection [45]. These methods provide useful foundations for threshold adaptation, latent-density estimation, and uncertainty-aware anomaly scoring, although their transfer to flight data requires domain-specific validation.

More directly related to aerial systems, self-supervised autoencoder-based representation learning has been applied to class-imbalanced UAV flight data, with wavelet analysis incorporated to improve fault-feature extraction. LSTM autoencoders combined with self-attention have also been investigated for unsupervised fault detection in aero-engine time-series data [46,47]. These studies extend reconstruction-based detection beyond generic time-series benchmarks, although their fault mechanisms, data-generation procedures, and evaluation settings remain task-specific.

To address the limitations of point-based reconstruction, a pattern-based attention recurrent autoencoder was proposed for anomaly detection in Ref. [48]. This framework captures temporal blocks of data rather than individual points, utilizing attention mechanisms to focus on relevant flight sequences. This approach effectively enhances the interpretability of anomalies in multivariate time series and provides a robust alternative to static thresholding in dynamic flight environments. Two key improvements to the basic LSTM-AE framework were further introduced in Ref. [49]. Savitzky–Golay filtering was used to remove random noise from the data, and the maximum information coefficient [50] was employed to select spatiotemporally strong correlated parameters. On the Thor Flight 69 dataset, the method achieved a true positive rate of 100% and simultaneously sup-

ported the recovery of anomalous data. In addition, to address the problem that overfitting in unsupervised reconstruction may result in excessively small reconstruction errors for anomalous samples, a multi-task learning framework was proposed in Ref. [51], in which semantic feature constraints were introduced into the bottleneck layer. This effectively improved the robustness of the model on the UAV altitude dataset, with the highest accuracy reaching 95.47%.

4.2.2. Regression-Based Detection Method Driven by Spatiotemporal Correlations

Such methods establish nonlinear mappings among flight parameters and evaluate anomalies based on the deviation between the actual system state and the model-predicted values. They provide strong capabilities for anomaly localization and data compensation, while also supporting the recovery of anomalous data.

A method incorporating artificial-neural-network-based correlation analysis was proposed in Ref. [52] to effectively capture nonlinear spatiotemporal correlations among parameters and overcome the limitation of the traditional Pearson correlation coefficient, which can only characterize linear relationships. Based on the correlation analysis results, the model input parameters are selected to avoid overfitting and excessive computational complexity caused by full-parameter input. On the self-developed Clouds fixed-wing UAV dataset, this method achieved 100% detection accuracy for bias anomalies, stationary anomalies, and point anomalies, and reduced the mean squared error (MSE) of anomaly recovery by 92.1% compared with baseline methods such as SVR.

The advantages of 1D CNN and LSTM were combined in Ref. [49], with the 1D CNN used to extract local features and the LSTM employed to capture long-term temporal dependencies, thereby comprehensively characterizing the spatiotemporal features of flight data. This approach enables the dual extraction of local morphological features and long-sequence dependencies. Combined with an exponentially weighted moving average filtering mechanism, it further suppresses false alarms caused by environmental noise and demonstrates superior overall performance to STC-LSTM and LSTM-RF on the Thor Flight series datasets.

4.3. Risk Probability Prediction

Based on flight-data-driven anomaly detection, LSTM-based models have successfully evolved from simple detection tasks into deep early warning applications involving takeoff risk assessment, approach stability prediction, and actuator fault localization through the introduction of statistical probability and uncertainty evaluation. To meet the safety requirements of civil aircraft and UAVs across different flight phases, existing studies have adopted diversified model architecture combinations to address the objectivity and real-time performance of risk representation, as well as the accuracy of anomaly localization.

4.3.1. Probabilistic Prediction and Assessment of Key Risks

For tire-overspeed risk during takeoff, Ref. [53] combined quantile analysis, bootstrap confidence intervals, and a two-layer LSTM to construct a five-level probabilistic risk-grading system, reporting $R^2 = 0.988$ within its QAR-based experimental setting. For unstable-approach prediction, Ref. [54] reported warning lead times of 30–60 s and a detection rate of 98.94% across 89 European airports. QAR-based risk assessment for approach and landing at high-altitude airports has also highlighted the need to incorporate environment-specific operating conditions and risk thresholds [55]. Physics-informed multimodal architectures have further been explored for proactive warning of vortex-ring-state risks in coaxial UAVs [56]. These findings provide task-specific evidence for probabilistic and continuous risk warning, but they should not be directly compared because the flight phases, datasets, risk definitions, and evaluation settings differ.

4.3.2. Robust Fault Monitoring Paradigm for UAV Sensors and Actuators

Given the high dynamics, susceptibility to environmental disturbances, and limited computational resources of UAV systems, researchers have proposed various lightweight and robustness-enhanced early warning frameworks.

An LSTM-based prediction–residual detection framework was proposed in Ref. [57], enabling the unsupervised identification of point anomalies in UAV sensors by constructing a 99% confidence interval for the residuals. Its label-free nature makes it highly suitable for small-sample and high-noise UAV flight scenarios. A joint strategy combining a dual-threshold method with LSTM was further proposed for gyroscope fault detection in [58], outperforming traditional models such as ARIMA and CNN. The dual-threshold strategy effectively balances detection sensitivity and false alarm rate through hierarchical decision-making, achieving a fault detection accuracy of over 94% on the BASIC dataset.

To address the challenge of large prediction fluctuations under complex flight conditions, an uncertainty-aware LSTM (UA-LSTM) method was developed in Ref. [59]. This method uses bidirectional phase-space reconstruction and GMM-clustering-based adaptive thresholds to dynamically characterize the prediction uncertainty of the model under different flight modes. It achieves an accuracy of 83.58% in detecting small-magnitude actuator bias faults, significantly outperforming conventional fixed-threshold LSTM. The method can adapt to variations in model prediction uncertainty under different flight conditions and substantially improves the detection sensitivity for small-magnitude faults.

4.4. Advanced Fusion Applications

As UAV application scenarios become increasingly complex, purely time-series-based anomaly detection can no longer satisfy the collaborative requirements of flight control. To address key bottlenecks in the aviation field, including environmental partial observability, scarcity of high-fidelity labeled data, and navigation failure in complex denied environments, researchers have deeply coupled the temporal feature extraction capability of LSTM with reinforcement learning, transfer learning, and multi-sensor fusion mechanisms, thereby constructing advanced perception frameworks with cross-domain adaptability and autonomous control capability. Deep reinforcement learning has become an important extension for drone perception–control coordination, with existing reviews highlighting its potential in trajectory planning, resource allocation, and autonomous control under partially observable environments [60].

4.4.1. Closed-Loop Optimization Paradigm Based on Perception–Control Coordination

To address the partial observability of flight states and communication resources in UAV-assisted communication scenarios, a deep reinforcement learning framework based on LSTM-derived temporal features was proposed in Ref. [61]. The framework predicts time-varying aerial channel quality, ground-sensor energy-harvesting patterns, and data-arrival probabilities, and subsequently feeds these representations into a deep deterministic policy gradient controller. DeFRA enables the joint optimization of UAV heading, velocity, and communication-resource allocation in continuous action spaces. Experimental results showed that the proposed framework reduced the packet-loss rate by more than 15% compared with existing reinforcement-learning schemes.

Complementary work on data-driven aircraft modeling and robust reinforcement-learning control with flight-test validation further emphasizes that control-oriented learning should be evaluated not only in simulation but also against real flight dynamics and operational disturbances [62].

Furthermore, to bridge the critical gap between temporal anomaly detection and active flight safety, recent hybrid approaches have pioneered the integration of LSTM

autoencoders directly with flight-control schemes for UAV fault diagnosis. A prominent framework proposed in the recent literature successfully combines conditional LSTM autoencoders (C-LSTM-AEs) with residual-based fault detection and isolation (FDI) within a closed-loop control system [63]. Unlike conventional open-loop detection systems that solely issue diagnostic alerts, this hybrid paradigm leverages the deep reconstruction capabilities of the C-LSTM-AEs to generate precise temporal residuals. These residuals are then utilized to actively isolate actuator or sensor faults and dynamically reconfigure the flight control laws in real time. This integration represents a significant methodological shift from passive state monitoring to robust fault-tolerant control, providing a crucial and practical connection between temporal sequence modeling, anomaly detection, and control-oriented aerial applications.

4.4.2. Cross-Domain Knowledge Transfer Paradigm Based on Multi-Source Integration

To address the industrial challenge that anomalous samples are extremely scarce in real flight environments, using abundant simulated data to guide real flight missions has become a research hotspot. A multi-source ensemble transfer learning framework, MSETL-AD, was developed in Ref. [64]. It quantifies the similarity between the source and target domains using dynamic time warping (DTW) [65] to suppress negative transfer effects, while incorporating an attention mechanism to enhance the extraction of critical temporal anomaly features. The framework transfers knowledge from five groups of simulated-domain data to real flight-test data, achieving accuracies of 91.90% and 93.02% in detecting two typical anomaly types, namely bias and drift, respectively. The principal contribution is the integration of DTW-based source-domain weighting, attention-enhanced feature extraction, and adaptive thresholding to reduce negative transfer under changing noise conditions.

4.4.3. Complex-Environment Navigation Enhancement Paradigm Based on Multimodal Perception

Maintaining autonomous flight navigation accuracy in GNSS-denied environments or environments with strong interference represents a frontier challenge for aviation safety. An LSTM-enhanced multi-sensor fusion navigation system was developed in Ref. [66], integrating LSTM with monocular visual odometry, FMCW-radar-based altitude estimation, and extended Kalman filtering. By learning the velocity drift patterns during periods when GNSS signals are available, the LSTM module provides high-accuracy correction values when the signals fail. This effectively mitigates the cumulative errors of visual odometry and the scale ambiguity of monocular ranging. Experimental results show that, under a GNSS outage lasting 113 s, the system can still reduce the lateral velocity error by 54.33%, demonstrating the robustness-enhancing effect of LSTM for navigation under complex physical constraints and multi-sensor fusion environments. This result provides task-specific evidence that learned velocity-error correction can improve navigation robustness during temporary GNSS outages.

5. A Multidimensional Evaluation Framework for Flight-Data-Driven LSTM-Family Models

As flight data mining evolves from single sequence modeling toward multimodal coupled architectures integrating temporal, spatial, and topological information, a single accuracy metric can no longer comprehensively reflect the engineering value of a model in complex aviation environments. To objectively and systematically evaluate the effectiveness of different LSTM variants and their hybrid architectures in aviation applications, and considering the fundamentally different tolerance boundaries of tasks such as trajectory prediction, anomaly detection, and risk assessment, this section extends the conventional

flat evaluation framework and constructs a bottom-up four-layer multidimensional comprehensive evaluation system, with highly customized implementation designs tailored to different application scenarios.

5.1. A Four-Layer Comprehensive Evaluation Foundation for Aeronautical Engineering

The proposed evaluation framework comprises four complementary layers: the data-algorithm layer, the model-performance layer, the physical-mechanism layer, and the system-engineering layer.

The proposed evaluation framework comprises four layers and ten dimensions. The data-algorithm layer evaluates spatiotemporal representation, robustness to non-stationary inputs, and attribute awareness. The model-performance layer covers task accuracy and accumulated spatial drift, generalization and out-of-distribution robustness, and uncertainty quantification, including prediction-interval reliability and Bayesian uncertainty estimation [67,68]. The physical-mechanism layer assesses consistency with aeronautical dynamics and kinematics, together with anomaly traceability supported by physics-informed constraints and feature-attribution techniques [69–71]. The system-engineering layer evaluates onboard size, weight, power, computational-resource requirements, and end-to-end real-time performance [72,73].

These four layers are complementary rather than independent. Data representation determines the quality of the learned features, model performance reflects task-specific predictive or diagnostic capability, physical consistency determines whether the outputs are operationally credible, and system-engineering indicators determine whether the model can operate on the intended aerial platform. The measurable indicators, units, preferred directions, normalization procedures, aggregation rules, and mandatory feasibility requirements of the ten dimensions are defined in Section 5.4 and Table 6.

5.2. Evaluation Orientation for Three Core Applications

Under the unified evaluation foundation, different application tasks exhibit significant differences in their priorities and tolerance boundaries across the four layers during practical deployment.

5.2.1. Prediction-Oriented Applications

Prediction-oriented applications include flight-trajectory forecasting, state estimation, delay prediction, and full-flight-phase variable extrapolation. Their evaluation should jointly consider pointwise prediction accuracy, accumulated spatial drift, robustness under unseen operating conditions, physical consistency, and deployment latency. Metrics such as RMSE, MAE, and R^2 characterize pointwise errors, whereas ADE and FDE quantify accumulated trajectory deviation. For safety-critical applications, prediction outputs should also be assessed against flight-envelope and kinematic constraints. When online deployment is claimed, the causal structure, input-window buffering, target hardware, and end-to-end latency should be explicitly reported [11,39]. The corresponding evaluation criteria are summarized in Table 2.

5.2.2. Anomaly Detection-Oriented Applications

Anomaly detection applications aim to identify sensor faults, actuator faults, abnormal maneuvers, and command-execution deviations from highly imbalanced flight data. Their evaluation should emphasize F1-score, AUC, false-alarm rate, missed-detection rate, minimum detectable fault magnitude, and robustness under variable trajectories. Because high overall accuracy may conceal poor minority-class performance, accuracy should not be used as the sole evaluation metric. Operationally useful anomaly detection systems should also provide fault-channel localization, distinguish genuine faults from maneuver-

induced deviations, and report the computational cost of continuous inference [6,49,59]. The corresponding evaluation criteria are summarized in Table 3.

Table 2. Evaluation criteria for prediction-oriented applications.

Evaluation Focus	Core Evaluation Metrics	Required Evidence
Pointwise accuracy and accumulated drift	RMSE, MAE, R^2 , ADE, FDE	Metric units, prediction horizon, and test-set definition
Generalization and OOD robustness	OOD performance loss, cross-validation standard deviation	Unseen weather, route, aircraft, or flight-phase conditions
Physical consistency	Flight-envelope violation rate, kinematic or aerodynamic residual	Explicit constraints and task-specific acceptance limits
Deployment feasibility	Model-only latency, end-to-end latency, memory, power, and operating rate	Target hardware, input-window length, and sampling or control-loop frequency

Table 3. Evaluation criteria for anomaly detection applications.

Evaluation Focus	Core Indicators	Required Evidence
Classification and false-alarm control	F1-Score, AUC, FAR, MDR	Class distribution, anomaly labels, threshold-selection method
Weak-fault sensitivity and robustness	Minimum detectable fault magnitude, performance under noise or trajectory changes	Fault type, magnitude, duration, and noise conditions
Traceability and recovery	Fault-channel localization accuracy, explanation fidelity, reconstruction or recovery error	Channel-level labels and attribution or recovery procedure
Continuous deployment	End-to-end latency, memory, power, operating rate	Target hardware and continuous monitoring frequency

5.2.3. Risk Assessment-Oriented Applications

Risk-assessment applications transform predicted states, detected anomalies, uncertainty information, and operational constraints into risk probabilities, warning levels, or intervention decisions. Their evaluation should therefore emphasize robustness to distribution shift, uncertainty calibration, warning lead time, warning continuity, and decision interpretability. PICP, normalized MPIW, calibration error, and the Brier score can be used to assess probabilistic reliability, while warning lead time and continuous alert-window duration reflect operational usefulness. Distribution-free calibration methods such as conformal prediction may further support uncertainty-aware monitoring under changing operating conditions [53,54,67,68,74,75]. The corresponding evaluation criteria are summarized in Table 4.

Table 4. Evaluation criteria for risk-assessment applications.

Evaluation Focus	Core Indicators	Required Evidence
Generalization and distribution shift	OOD performance loss, cross-validation variability	Flight phase, environment, aircraft, and data-partition conditions
Uncertainty calibration	PICP, normalized MPIW, ECE, Brier score	Nominal coverage level and calibration procedure
Warning timeliness and continuity	Warning lead time, continuous alert-window ratio	Risk-event definition and operational intervention horizon
Decision interpretability	Risk-factor localization accuracy, explanation fidelity or stability	Attribution method and domain-expert validation

The application-specific criteria summarized in Tables 2–4 are subsequently complemented by deployment-evidence analysis and common operational scoring rules.

5.3. Reported Deployment Evidence and Reporting Gaps

Although many flight-data studies describe their models as lightweight, real-time, or potentially suitable for onboard use, the reported engineering evidence is incomplete and heterogeneous. To distinguish demonstrated deployment capability from conceptual deployability, Table 5 summarizes only the resource- and timing-related information explicitly reported in representative studies. No missing deployment values were estimated from network descriptions or model configurations. Information absent from the original publications is marked as “NR”. The comparison distinguishes offline algorithmic evaluation, offboard real-time execution, simulation-based closed-loop validation, and direct onboard implementation.

Table 5 reveals a substantial reporting gap between algorithmic research and engineering deployment. Excluding the two comparative non-LSTM alternatives [42,76], 14 of the 16 representative LSTM-family studies (87.5%) did not report any quantitative deployment-related metric. Among the selected LSTM-family studies, at least one quantitative indicator related to execution time, computational complexity, model scale, or hardware configuration was reported in only two studies [11,77], accounting for 12.5% of the selected studies. Moreover, only [77] provided direct quantitative onboard validation, representing 6.25% of this representative subset.

These percentages describe only the representative studies summarized in Table 5 and should not be extrapolated to all publications identified during the literature search. Nevertheless, the pattern is consistent: most existing studies emphasize prediction, detection, or warning performance while omitting serialized model size, runtime memory, numerical precision, FLOPs or MACs, power consumption, and complete target-hardware configurations. Even when terms such as “lightweight,” “real-time,” or “online” are used, they are frequently unsupported by measurements on the intended onboard platform.

The absence of deployment information should not be interpreted as evidence that the corresponding models are inherently unsuitable for onboard implementation. Instead, it indicates that their deployment feasibility cannot be independently assessed from the published information. Accordingly, the resource-aware framework proposed in this review serves two complementary purposes: evaluating the currently available deployment evidence and defining a standardized reporting checklist for future flight-data studies. At a minimum, future studies should report trainable parameter count, serialized model size, peak runtime memory, FLOPs or MACs per inference, numerical precision, target hardware, input-window length, batch size, model-only inference latency, end-to-end latency, throughput or operating rate, average power, and energy per inference.

5.4. Operationalization and Illustrative Evidence Profiling of the Evaluation Framework

To ensure reproducible model evaluation, quantitative scoring is performed separately for trajectory prediction, anomaly detection, and risk assessment (Table 6). Models are included in the same comparison group only when they use sufficiently comparable datasets, data partitions, prediction horizons or anomaly definitions, baseline configurations, metric definitions, and hardware conditions. Cross-study numerical results that do not satisfy these conditions are used only for descriptive evidence synthesis and are not included in quantitative ranking.

Table 5. Deployment-relevant evidence reported in representative flight-data studies.

Ref.	Model and Application	Validation Mode	Parameters	Model Size	Runtime Memory	FLOPs/MACs	Precision	Hardware	Inference Latency	Power/Energy	Operating Rate
[2]	LSTM for flight-trajectory prediction	Offline evaluation	NR	NR	NR	NR	NR	NR	NR	NR	NR
[6]	Multirate-aware LSTM for UAV anomaly detection	Offline evaluation	NR	NR	NR	NR	NR	NR	NR	NR	NR
[8]	DMPSO-LSTM for full-flight fuel-consumption prediction	Offline QAR-based evaluation	NR	NR	NR	NR	NR	NR	NR	NR	NR
[10]	mRMR-LSTM for landing-attitude warning	Offline QAR-based warning evaluation	NR	NR	NR	NR	NR	NR	NR; 5 s warning lead time reported separately	NR	NR
[11]	EC-Bessel-BiLSTM for UAV path prediction	Offboard real-time evaluation	NR	NR	NR	Analytical FLOP expression reported; final value NR	NR	Ground station: Intel Xeon E5-2678 v3 CPU and NVIDIA GeForce RTX 2080 Ti GPU	53.1–59.8 ms per prediction	NR	NR
[15]	CNN-LSTM for aircraft 4D trajectory prediction	Offline algorithmic evaluation	NR	NR	NR	NR	NR	NR	NR	NR	NR
[38]	Att-Conv-LSTM for flight-delay prediction	Ground-based offline analysis	NR	NR	NR	NR	NR	NR	NR	NR	NR

Table 5. Cont.

Ref.	Model and Application	Validation Mode	Parameters	Model Size	Runtime Memory	FLOPs/MACs	Precision	Hardware	Inference Latency	Power/Energy	Operating Rate
[39]	Constrained LSTM for aircraft 4D trajectory prediction	Offline algorithmic evaluation	NR	NR	NR	NR	NR	NR	NR	NR	NR
[41]	Incremental RLSTM for UAV trajectory prediction	Online algorithm updating; target hardware NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
[42]	Sta_POS NARX for drone-motion prediction	Offline evaluation	NR	NR	NR	NR	NR	NR	NR; 3–5× training speedup but not inference latency	NR	NR
[49]	CNN-LSTM-based spatiotemporal anomaly detection and recovery	Offline dataset evaluation	NR	NR	NR	NR	NR	NR	NR	NR	NR
[76]	GBDT-Informer for long-sequence landing-risk prediction	Offline algorithmic evaluation	NR	NR	NR	NR	NR	NR	NR	NR	NR
[53]	Quantile-Bootstrap-LSTM for tire-overspeed risk assessment	Offline QAR-based evaluation	NR	NR	NR	NR	NR	NR	NR	NR	NR

Table 5. Cont.

Ref.	Model and Application	Validation Mode	Parameters	Model Size	Runtime Memory	FLOPs/MACs	Precision	Hardware	Inference Latency	Power/Energy	Operating Rate
[59]	UA-LSTM for small actuator-fault detection	Offline algorithmic evaluation	NR	NR	NR	NR	NR	NR	NR	NR	NR
[61]	LSTM-enhanced DRL for UAV control and resource allocation	Simulation/testbed evaluation; onboard execution NR	NR	NR	NR	NR	NR	NR	NR	NR	NR
[64]	MSETL-AD for simulation-to-real anomaly detection	Transfer-learning evaluation	NR	NR	NR	NR	NR	NR	NR	NR	NR
[63]	C-LSTM-AE for fault diagnosis and FDI	Simulation-based closed-loop validation	NR	NR	NR	NR	NR	Simulation platform; detailed deployment hardware NR	NR	NR	NR
[77]	Lightweight LSTM for onboard wind estimation	Direct onboard implementation	1281	NR	NR	NR	NR	Onboard Raspberry Pi with Pixhawk/MAVLink interface	4 ms per LSTM inference cycle	NR	50 Hz complete control loop

Note: NR indicates that the corresponding information was not reported in the original publication. Warning lead time, model inference latency, and end-to-end control-loop time are distinct quantities and are reported separately. A reported training-speed improvement is not treated as inference-latency evidence. Execution on a workstation or ground station is classified as offboard evaluation and should not be interpreted as direct onboard deployment. Because the reviewed studies differ in application tasks, input-window lengths, hardware environments, numerical precision, and timing definitions, the table represents the level and completeness of reported deployment evidence rather than a direct hardware-performance ranking. References [42,76] are comparative non-LSTM or nonstandard recurrent alternatives discussed in the review. They are retained to illustrate the broader reporting status but are excluded from the LSTM-family reporting-rate calculation.

Table 6. Operational definitions of the four-layer and ten-dimensional evaluation framework.

Layer	Dimension	Measurable Indicators	Unit	Preferred Direction
Data-Algorithm	D1. Spatiotemporal representation	Within-study improvement over causal LSTM; ablation gain from CNN/GCN modules	Improvement over baseline, %	Higher
Data-Algorithm	D2. Robustness to non-stationary inputs	Relative degradation under noise, temporal shift, or OOD data	Performance degradation, %	Lower
Data-Algorithm	D3. Attribute awareness	Feature-localization precision; attention/feature-selection ablation gain	Localization accuracy or ablation gain, %	Higher
Model-Performance	D4. Task accuracy and accumulated drift	RMSE, MAE, ADE, FDE, F1-score, AUC, FAR, MDR, or (R ²), according to task	M, m/s, %	Task-dependent
Model-Performance	D5. Generalization and OOD robustness	OOD performance loss; cross-validation standard deviation	OOD performance loss, %; cross-validation SD	Lower
Model-Performance	D6. Uncertainty quantification	PICP, normalized MPIW, calibration error or Brier score	PICP, %; normalized MPIW, ECE, Brier score	Target-dependent
Physical-Mechanism	D7. Physical consistency	Flight-envelope violation rate; kinematic or aerodynamic residual	Physical-constraint violation rate, %	Lower
Physical-Mechanism	D8. Interpretability and traceability	Fault-channel localization accuracy; explanation fidelity or stability	Localization accuracy, %; fidelity/stability score	Higher
System-Engineering	D9. SWaP and resource cost	Parameters, serialized model size, runtime memory, FLOPs/MACs, power, and energy	Parameters, MB, FLOPs/MACs, W, mJ/inference	Lower
System-Engineering	D10. Real-time performance	Model-only latency, end-to-end latency, throughput and operating rate	ms/inference, ms/end-to-endHz, samples/s	Task-dependent

5.4.1. Normalization and Aggregation Rules

Quantitative scoring is performed only within a predefined comparable group. A comparable group consists of models evaluated using sufficiently consistent datasets, data partitions, prediction horizons or anomaly definitions, baseline configurations, metric definitions, and hardware conditions. Results from heterogeneous experimental settings are retained for descriptive evidence synthesis but are not included in the same numerical ranking.

Because the indicators have different units and preferred directions, each applicable indicator is first normalized to the interval [0,1]. For an indicator for which a higher value represents better performance, the normalized score is calculated as

$$s_{ij} = \frac{x_{ij} - x_j^{\min}}{x_j^{\max} - x_j^{\min}}, \quad (4)$$

where x_{ij} denotes the reported value of indicator j for model i , and $x_j^{\min}$ and $x_j^{\max}$ denote the minimum and maximum values of indicator j , respectively, within the predefined comparable group.

For an indicator for which a lower value represents better performance, the normalized score is calculated as

$$s_{ij} = \frac{x_j^{\max} - x_{ij}}{x_j^{\max} - x_j^{\min}}. \quad (5)$$

The normalized value is subsequently converted to a 0–100 score:

$$q_{ij} = 100s_{ij}. \quad (6)$$

When all models in a comparable group have the same value for a given indicator, that indicator has no discriminative capability and is excluded from the aggregate score to avoid division by zero.

The overall score of model i is calculated as the weighted average of its applicable and reported indicator scores:

$$Q_i = \frac{\sum_{j \in A_i} w_j q_{ij}}{\sum_{j \in A_i} w_j}, \quad (7)$$

where A_i denotes the set of applicable and quantitatively reported indicators for model i , and w_j denotes the weight assigned to indicator j . To minimize subjective assumptions and ensure transparent reproducibility, equal weights are assigned to all applicable indicators, that is, $w_j = 1$. Therefore, the aggregate score represents the arithmetic mean of the available applicable indicator scores.

Weight-sensitivity analysis is not performed because the present framework is intended as a transparent rule-based evidence-synthesis and model-screening procedure rather than a statistically learned ranking system. Moreover, the aggregate scores are interpreted only within each predefined comparable group and should not be used to establish a universal ranking across trajectory-prediction, anomaly detection, and risk-assessment studies.

5.4.2. Treatment of Missing and Non-Applicable Information

Indicators that are irrelevant to a particular task are marked as “NA” (not applicable) and are excluded from both the numerator and denominator of the aggregate score. Information that is applicable but not reported in the original publication is marked as “NR” (not reported). NR values are not estimated from model descriptions, are not assigned a score of zero, and are excluded from numerical aggregation.

To make the completeness of the available evidence explicit, an evidence-coverage ratio is calculated as

$$C_i = \frac{N_i^{\text{reported}}}{N_i^{\text{applicable}}}, \quad (8)$$

where N_i^{reported} is the number of applicable indicators quantitatively reported for model i , and $N_i^{\text{applicable}}$ is the total number of indicators applicable to that model. An aggregate score is reported only when at least 70% of the applicable indicators are available. When the evidence-coverage ratio is below this threshold, only a multidimensional evidence profile is presented, and no overall ranking is assigned.

5.4.3. Mandatory Feasibility Requirements

Physical consistency, onboard resource feasibility, and real-time performance are treated as non-compensatory requirements when they are applicable to the target task. A high score in prediction or detection accuracy cannot compensate for the violation of a mandatory physical or engineering requirement.

Specifically, a model is considered operationally feasible only when:

- (1) Its physical-constraint violation rate satisfies the predefined task-specific tolerance;
- (2) Its memory, computational, and power requirements do not exceed the capacity of the target onboard platform;

- (3) Its end-to-end latency does not exceed the sampling or control-loop period of the target system.

The real-time requirement is expressed as

$$t_{\text{end-to-end}} \leq \frac{1000}{f_{\text{operation}}} \text{ms}, \quad (9)$$

where $f_{\text{operation}}$ is the required sampling, monitoring, or control-loop frequency in hertz. Failure to satisfy any mandatory requirement results in the model being classified as operationally infeasible for the corresponding application, regardless of its aggregate score. When a mandatory indicator is not reported, operational feasibility is classified as “insufficient evidence” rather than “pass” or “fail”.

Not every indicator is equally applicable to every application category. Trajectory-prediction models are primarily evaluated using pointwise prediction errors, accumulated displacement errors, generalization performance, and physical consistency. Anomaly detection models emphasize F1-score, AUC, false-alarm rate, missed-detection rate, fault sensitivity, and anomaly localization capability. Risk-assessment models emphasize uncertainty calibration, warning lead time, warning continuity, and decision interpretability. System-engineering indicators are included only when deployment or real-time feasibility is claimed. Consequently, the framework produces task-specific multidimensional profiles and, where sufficient comparable evidence is available, within-group aggregate scores. It does not generate a single universal ranking across all aerial-system applications.

Table 7 provides illustrative evidence profiles showing how the proposed framework distinguishes task performance, deployment evidence, feasibility requirements, and missing information. Because the selected models address different tasks and were evaluated under heterogeneous conditions, they are not combined into a universal numerical ranking.

Table 7. Illustrative evidence profiles of representative studies using the proposed evaluation framework.

Ref.	Model and Task	Quantitatively Reported Evidence	Quantitatively Supported Dimensions	Applicable but Unreported/Partially Reported Dimensions	Framework Interpretation
[41]	Incremental RLSTM; trajectory prediction	Mean 3D prediction error of 6.25 m and within-study comparisons with MLP, RNN, LSTM, and BiLSTM	D4	D2, D5, D7, D9, D10	Supports a within-study accuracy comparison. Robustness, physical consistency, and deployment feasibility cannot be assessed from the reported evidence.
[59]	UA-LSTM; UAV actuator-fault detection	Small-bias fault-detection accuracy, fixed-threshold baseline comparison, and adaptive-threshold results under different flight conditions	D4	D2, D5, D6, D8, D9, D10	Provides task-specific detection evidence. Uncertainty-aware thresholding is demonstrated, but uncertainty calibration, OOD robustness, traceability, and deployment feasibility remain insufficiently reported.
[77]	Lightweight LSTM; onboard wind estimation	1281 parameters, Raspberry Pi implementation, 4 ms model inference, and a 50 Hz end-to-end control loop	D4, D9 (partial), D10	D5, D7; model size, runtime memory, FLOPs/MACs, numerical precision, power, and energy within D9	Provides direct onboard performance and timing evidence. However, generalization, physical-consistency evidence, and a complete SWaP profile remain unavailable.

Note: “Quantitatively supported dimensions” include only dimensions for which the original study reported at least one indicator consistent with the operational definitions in Table 6. A dimension may be conceptually addressed by a study but classified as partially reported when the required quantitative indicator is unavailable. D9 is marked as partial for Ref. [77] because parameter count was reported, whereas model size, runtime memory, FLOPs/MACs, numerical precision, power, and energy were not reported. The three studies were evaluated under heterogeneous tasks and conditions and are therefore presented as evidence profiles rather than as a numerical ranking.

6. Discussion and Limitations

6.1. Evidence Synthesis and Cross-Study Comparability

The reviewed evidence supports conditional rather than universal conclusions regarding the relative advantages of LSTM-family architectures. Temporal, convolutional, graph-based, reconstruction-aware, and uncertainty-aware models were generally evaluated using different datasets, aircraft platforms, flight phases, prediction horizons, anomaly definitions, data partitions, and baseline configurations. Consequently, the numerical results summarized in the tables should not be interpreted as a unified ranking of model performance. Comparative conclusions are reliable only when the models were evaluated within the same study using identical datasets, data partitions, prediction settings, and evaluation metrics.

Within these constraints, several recurring patterns can be identified. Convolution-enhanced LSTM models tend to be useful when local temporal or grid-like patterns are explicitly represented, whereas graph-based architectures are more relevant when airports, sectors, routes, or aircraft form an explicit relational network. However, the reported benefits depend strongly on feature organization, graph construction, prediction horizon, and baseline selection. Similarly, physics-constrained models may reduce physically implausible outputs, but the available evidence remains limited to specific aircraft models, flight phases, and constraint definitions. These observations should therefore be regarded as task- and setting-dependent evidence rather than general guarantees of model superiority.

Evidence for resource-aware onboard deployment is considerably less complete than evidence for algorithmic accuracy. As summarized in Table 5, most representative studies do not report sufficient information on parameter count, serialized model size, runtime memory, FLOPs or MACs, numerical precision, end-to-end latency, power consumption, operating rate, or target onboard hardware. Consequently, terms such as “lightweight”, “real-time”, “online”, and “edge-deployable” are interpreted cautiously unless they are supported by quantitative measurements on a clearly identified platform. Feature reduction, pruning, quantization, low-rank factorization, FPGA implementation, and DSP acceleration should therefore be regarded as potential deployment strategies rather than universally validated solutions.

6.2. Limitations of This Review

This review has several limitations. First, the reviewed studies exhibit substantial heterogeneity in data sources, aircraft platforms, flight phases, prediction horizons, anomaly definitions, data-partition strategies, baseline models, and evaluation metrics. This heterogeneity prevents direct quantitative comparison and does not support a formal meta-analysis of the reported results. Second, many studies incompletely report experimental and deployment conditions, including dataset size, train–validation–test partitions, model parameters, memory consumption, numerical precision, inference latency, and energy use. Missing information limits the reproducibility of both model-performance and onboard-deployment assessments. In addition, Table 5 contains a substantial number of NR entries because the original studies did not report the requested engineering indicators. The table therefore supports an assessment of reporting completeness and deployment-validation level, but it does not permit a normalized ranking of hardware efficiency across models. The representative-study selection also means that the reported proportions should not be interpreted as prevalence estimates for the entire body of LSTM-related aviation literature. Third, the proposed multidimensional evaluation framework is primarily a structured methodological synthesis rather than a statistically validated scoring system, and its practical weighting may vary across prediction, anomaly detection, and risk-assessment tasks. Finally, publication bias, language restrictions, and the limited availability of publicly

accessible aviation datasets may affect the coverage of the reviewed evidence. Although open resources such as OpenSky, TrajAir, and TartanAviation provide valuable foundations for reproducible research, they differ substantially in airspace type, sensing modality, geographical coverage, sampling conditions, and annotation granularity [3–5]. Therefore, they do not yet constitute a unified benchmark for comparing flight-data-driven models. These limitations should be considered when interpreting the conclusions and model-selection guidance presented in this review.

7. Conclusions and Future Directions

As the current aerospace industry and unmanned aerial vehicle sector move toward higher levels of autonomy and swarm-based operation [78], ensuring flight safety under complex maneuvering trajectories and highly dynamic environments has become a critical issue in this field. This review examined the architectural evolution, application paradigms, evaluation requirements, and deployment evidence of flight-data-driven LSTM-family models for aerial systems.

Based on the study-specific evidence summarized in the structured evidence tables, three main observations can be made. First, the value of an LSTM-family architecture depends on the structure of the data and the target application. Standard LSTM remains a practical baseline for causal temporal modeling, convolution-enhanced architectures can improve local spatiotemporal representation, and graph-based architectures are suitable when explicit relational structures among aircraft, airports, routes, or sectors are available. However, the effectiveness of these extensions depends on prediction settings, graph construction, data availability, and computational constraints.

Second, trajectory prediction, anomaly detection, and risk assessment require differentiated evaluation criteria. Trajectory-prediction studies should jointly report pointwise errors, accumulated spatial drift, and physical consistency. Anomaly detection studies should emphasize false-alarm control, missed detections, class imbalance, anomaly localization, and recovery capability. Risk-assessment studies should further evaluate uncertainty calibration, warning lead time, and decision interpretability. Because the reviewed studies use heterogeneous experimental conditions, these criteria provide task-specific assessment guidance rather than a direct ranking across publications.

Third, feature reduction, model compression, and hardware-aware acceleration offer potential routes for narrowing the gap between algorithmic validation and onboard implementation. However, the deployment evidence summarized in Table 5 remains incomplete. Most representative studies do not simultaneously report trainable parameter count, serialized model size, runtime memory, FLOPs or MACs, numerical precision, target hardware, inference latency, operating rate, and power or energy consumption. Only a limited subset provides quantitative execution or model-scale information, and direct onboard validation is confined to individual cases. Therefore, no single compression or hardware-acceleration strategy can currently be regarded as a universally established pathway for flight-safe onboard deployment.

Future research should move beyond the isolated optimization of individual architectures and focus on reliable, transferable, and hardware-aware flight intelligence. First, causal and streaming LSTM-family models should be developed for sample-by-sample onboard inference without relying on complete future sequence windows. Second, cross-aircraft, cross-route, and cross-airport generalization should be strengthened through transfer learning, domain adaptation, and explicit evaluation under distribution shifts and out-of-distribution operating conditions. Third, graph-based models require transferable and dynamically updated graph-construction mechanisms capable of representing time-varying interactions without introducing prohibitive computational costs. Fourth,

uncertainty estimates should be systematically calibrated across different flight phases, environmental conditions, and aircraft platforms so that confidence boundaries can support operational decision-making. Physics-informed constraints and causal interpretability should also be integrated to distinguish genuine system faults from pilot-induced maneuvers and environmental disturbances. Finally, closer model–hardware co-design, standardized flight datasets, and unified benchmark protocols will be essential for advancing LSTM-family models toward trustworthy onboard applications. Future studies should transparently report trainable parameters, serialized model size, peak runtime memory, FLOPs or MACs, numerical precision, target hardware, input-window length, batch size, model-only and end-to-end latency, operating rate, average power, and energy per inference. Standardized reporting of these indicators will enable deployment feasibility to be independently assessed and will support meaningful comparison across models evaluated on comparable aerial platforms.

Author Contributions: Conceptualization, F.W. and T.L.; resources, F.W., T.L., Y.W., Y.Z., Z.W., and H.H.; writing—original draft preparation, F.W. and T.L.; writing—review and editing, F.W., T.L., Y.W., Y.Z., Z.W., and H.H.; supervision, F.W. All authors have read and agreed to the published version of the manuscript.

Funding: This work was partially supported by the Research on the Construction of Think Tanks for Civil Aviation Major Emergency Transportation and National Defense Transport Safety Operation Assurance under grant No. YS2025001, Sichuan Provincial Engineering Research Center of Smart Operation and Maintenance of Civil Aviation Airports under grant No. JCZX2024ZZ16, and the Fundamental Research Funds for the Central Universities under grant Nos. 26CAFUC01008, 26CAFUC01010, 26CAFUC03048, 26CAFUC03077, 25CAFUC03036, 25CAFUC03037, and 25CAFUC09010.

Data Availability Statement: No new data were created or analyzed in this study. Data sharing is not applicable to this article.

Conflicts of Interest: The authors declare no conflicts of interest.

References

1. Yang, L.; Li, S.; Zhang, Y.; Zhu, C.; Liao, Z. Deep learning-assisted unmanned aerial vehicle flight data anomaly detection: A review. *IEEE Sens. J.* **2024**, *24*, 31681–31695. [\[CrossRef\]](#)
2. Shi, Z.; Xu, M.; Pan, Q.; Yan, B.; Zhang, H. LSTM-based flight trajectory prediction. In Proceedings of the 2018 International Joint Conference on Neural Networks (IJCNN), Rio de Janeiro, Brazil, 8–13 July 2018; pp. 1–8.
3. Strohmeier, M.; Martinovic, I.; Fuchs, M.; Schäfer, M.; Lenders, V. Opensky: A swiss army knife for air traffic security research. In Proceedings of the 2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC), Prague, Czech Republic, 13–17 September 2015; pp. 4A1-1–4A1-14.
4. Patrikar, J.; Moon, B.; Oh, J.; Scherer, S. Predicting like a pilot: Dataset and method to predict socially-aware aircraft trajectories in non-towered terminal airspace. In Proceedings of the 2022 International Conference on Robotics and Automation (ICRA), Philadelphia, PA, USA, 23–27 May 2022; pp. 2525–2531.
5. Patrikar, J.; Dantas, J.; Moon, B.; Hamidi, M.; Ghosh, S.; Keetha, N.; Higgins, I.; Chandak, A.; Yoneyama, T.; Scherer, S. Image, speech, and ADS-B trajectory datasets for terminal airspace operations. *Sci. Data* **2025**, *12*, 468. [\[CrossRef\]](#) [\[PubMed\]](#)
6. Lu, H.; Wang, Z.; Shi, Y. Unmanned aerial vehicle flight data anomaly detection based on multirate-aware LSTM. *IEEE Trans. Instrum. Meas.* **2024**, *73*, 1–13. [\[CrossRef\]](#)
7. Saripalli, H.H.B.; Saripalli, J.L.; Lagos, L.; Upadhyay, H. LSTM-VAE for Temporal Anomaly Detection in Drone Trajectory Analysis: A Comparative Study for Critical Infrastructure Protection. *Future Internet* **2026**, *18*, 301. [\[CrossRef\]](#)
8. Xiong, J.; Zou, C.; Wan, Y.; Sun, Y.; Yu, G. Fuel consumption prediction for full flight phases toward sustainable aviation: A DMPISO-LSTM model using quick access recorder (QAR) data. *Sustainability* **2025**, *17*, 3358. [\[CrossRef\]](#)
9. Torres, M.E.; Colominas, M.A.; Schlotthauer, G.; Flandrin, P. A complete ensemble empirical mode decomposition with adaptive noise. In Proceedings of the 2011 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP), Prague, Czech Republic, 22–27 May 2011; pp. 4144–4147.
10. Lu, F.; Jing, T.; Xie, C.; Chen, H. Civil Aircraft Landing Attitude Ultra-Limit Warning System Based on mRMR-LSTM. *Aerospace* **2025**, *12*, 581. [\[CrossRef\]](#)

11. Chen, S.; Chen, B.; Shu, P.; Wang, Z.; Chen, C. Real-time unmanned aerial vehicle flight path prediction using a bi-directional long short-term memory network with error compensation. *J. Comput. Des. Eng.* **2023**, *10*, 16–35.
12. Li, Q.; Guan, X.; Liu, J. A CNN-LSTM framework for flight delay prediction. *Expert Syst. Appl.* **2023**, *227*, 120287. [[CrossRef](#)]
13. Hao, Q.; Zhang, J.; Jing, T.; Wang, W. Flight trajectory prediction using an enhanced cnn-lstm network. *arXiv* **2024**, arXiv:2404.19218.
14. Shi, X.; Chen, Z.; Wang, H.; Yeung, D.Y.; Wong, W.K.; Woo, W.C. Convolutional LSTM network: A machine learning approach for precipitation nowcasting. *Adv. Neural Inf. Process. Syst.* **2015**, *28*. [[CrossRef](#)]
15. Ma, L.; Tian, S. A hybrid CNN-LSTM model for aircraft 4D trajectory prediction. *IEEE Access* **2020**, *8*, 134668–134680. [[CrossRef](#)]
16. Pang, Y.; Xu, N.; Liu, Y. Aircraft trajectory prediction using LSTM neural network with embedded convolutional layer. In *Proceedings of the Annual Conference of the PHM Society*; PHM Society: Scottsdale, AZ, USA, 2019; Volume 11, pp. 1–8.
17. Sutskever, I.; Vinyals, O.; Le, Q.V. Sequence to sequence learning with neural networks. *Adv. Neural Inf. Process. Syst.* **2014**, *27*. Available online: <https://dl.acm.org/doi/10.5555/2969033.2969173> (accessed on 30 July 2026).
18. Yang, L.; Li, S.; Zhu, C.; Liu, J.; Zhang, A. Temporal convolutional kolmogorov-arnold networks-based multivariate regression for unmanned aerial vehicle flight data anomaly detection and recovery. *IEEE Trans. Instrum. Meas.* **2025**, *74*, 2538117. [[CrossRef](#)]
19. Cai, K.; Li, Y.; Zhu, Y.; Fang, Q.; Yang, Y.; Du, W. A geographical and operational deep graph convolutional approach for flight delay prediction. *Chin. J. Aeronaut.* **2023**, *36*, 357–367. [[CrossRef](#)]
20. Bao, J.; Yang, Z.; Zeng, W. Graph to sequence learning with attention mechanism for network-wide multi-step-ahead flight delay prediction. *Transp. Res. Part C Emerg. Technol.* **2021**, *130*, 103323. [[CrossRef](#)]
21. Zheng, H.; Wang, Z.; Zheng, C.; Wang, Y.; Fan, X.; Cong, W.; Hu, M. A graph multi-attention network for predicting airport delays. *Transp. Res. Part E Logist. Transp. Rev.* **2024**, *181*, 103375. [[CrossRef](#)]
22. Zhang, Y.; Xu, S.; Zhang, L.; Jiang, W.; Alam, S.; Xue, D. Short-term multi-step-ahead sector-based traffic flow prediction based on the attention-enhanced graph convolutional LSTM network (AGC-LSTM). *Neural Comput. Appl.* **2025**, *37*, 14869–14888.
23. Kipf, T.N.; Welling, M. Semi-supervised classification with graph convolutional networks. *arXiv* **2016**, arXiv:1609.02907.
24. Veličković, P.; Cucurull, G.; Casanova, A.; Romero, A.; Lio, P.; Bengio, Y. Graph attention networks. *arXiv* **2017**, arXiv:1710.10903.
25. Yuan, Y.; Wang, Y.; Lai, C.S. Multi-attribute data-driven flight departure delay prediction for airport system using deep learning method. *Aerospace* **2025**, *12*, 246. [[CrossRef](#)]
26. Wu, Y.; Yang, H.; Lin, Y.; Liu, H. Spatiotemporal propagation learning for network-wide flight delay prediction. *IEEE Trans. Knowl. Data Eng.* **2023**, *36*, 386–400.
27. Wu, Z.; Pan, S.; Long, G.; Jiang, J.; Zhang, C. Graph wavenet for deep spatiotemporal graph modeling. *arXiv* **2019**, arXiv:1906.00121.
28. Zhou, H.; Zhang, S.; Peng, J.; Li, J.; Xiong, H.; Zhang, W. Informer: Beyond efficient transformer for long sequence time-series forecasting. In *Proceedings of the AAAI Conference on Artificial Intelligence*, Virtual, 2–9 February 2021; Volume 35, pp. 11106–11115. [[CrossRef](#)]
29. Wu, H.; Xu, J.; Wang, J.; Long, M. Autoformer: Decomposition transformers with auto-correlation for long-term series forecasting. *Adv. Neural Inf. Process. Syst.* **2021**, *34*, 22419–22430.
30. Fan, Y.; Liu, J.; Ye, H.; Lyu, Z. TA-LSTM: A time and attribute aware LSTM for deep flight track clustering. *IEEE Trans. Aerosp. Electron. Syst.* **2023**, *59*, 7047–7060.
31. Kingma, D.P.; Welling, M. Auto-encoding variational bayes. *arXiv* **2013**, arXiv:1312.6114.
32. Han, S.; Mao, H.; Dally, W.J. Deep compression: Compressing deep neural networks with pruning, trained quantization and huffman coding. *arXiv* **2015**, arXiv:1510.00149.
33. Jacob, B.; Kligys, S.; Chen, B.; Zhu, M.; Tang, M.; Howard, A.; Adam, H.; Kalenichenko, D. Quantization and training of neural networks for efficient integer-arithmetic-only inference. In *Proceedings of the IEEE Conference on Computer Vision and Pattern Recognition*, Salt Lake City, UT, USA, 18–22 June 2018; pp. 2704–2713.
34. Kusupati, A.; Singh, M.; Bhatia, K.; Kumar, A.; Jain, P.; Varma, M. Fastgrnn: A fast, accurate, stable and tiny kilobyte sized gated recurrent neural network. *Adv. Neural Inf. Process. Syst.* **2018**, *31*. [[CrossRef](#)]
35. Han, S.; Kang, J.; Mao, H.; Hu, Y.; Li, X.; Li, Y.; Xie, D.; Luo, H.; Yao, S.; Wang, Y.; et al. ESE: Efficient speech recognition engine with sparse lstm on fpga. In *Proceedings of the 2017 ACM/SIGDA International Symposium on Field-Programmable Gate Arrays*, Monterey, CA, USA, 22–24 February 2017; pp. 75–84.
36. Gao, C.; Rios-Navarro, A.; Chen, X.; Liu, S.C.; Delbruck, T. EdgeDRNN: Recurrent neural network accelerator for edge inference. *IEEE J. Emerg. Sel. Top. Circuits Syst.* **2020**, *10*, 419–432. [[CrossRef](#)]
37. Figuet, B.; Krauth, T.; Barry, S. Generative Short-Term Aircraft Trajectory Prediction with Conditional Flow Matching. *J. Open Aviat. Sci.* **2026**, *4*. [[CrossRef](#)]
38. Qu, J.; Xiao, M.; Yang, L.; Xie, W. Flight delay regression prediction model based on Att-Conv-LSTM. *Entropy* **2023**, *25*, 770. [[CrossRef](#)] [[PubMed](#)]

39. Shi, Z.; Xu, M.; Pan, Q. 4-D flight trajectory prediction with constrained LSTM network. *IEEE Trans. Intell. Transp. Syst.* **2020**, *22*, 7242–7255.
40. Ester, M.; Kriegel, H.P.; Sander, J.; Xu, X. A density-based algorithm for discovering clusters in large spatial databases with noise. In Proceedings of the kdd, Portland, OR, USA, 2–4 August 1996; Volume 96, pp. 226–231.
41. Zhang, Y.; Jia, Z.; Dong, C.; Liu, Y.; Zhang, L.; Wu, Q. Recurrent LSTM-based UAV trajectory prediction with ADS-B information. In Proceedings of the GLOBECOM 2022–2022 IEEE Global Communications Conference, Rio de Janeiro, Brazil, 4–8 December 2022; pp. 1–6.
42. Dong, S.; Das, S.; Townley, S. Drone motion prediction from flight data: A nonlinear time series approach. *Syst. Sci. Control Eng.* **2024**, *12*, 2409098. [[CrossRef](#)]
43. Hundman, K.; Constantinou, V.; Laporte, C.; Colwell, I.; Soderstrom, T. Detecting spacecraft anomalies using lstms and nonparametric dynamic thresholding. In Proceedings of the 24th ACM SIGKDD International Conference on Knowledge Discovery & Data Mining, London, UK, 19–23 August 2018; pp. 387–395.
44. Zong, B.; Song, Q.; Min, M.R.; Cheng, W.; Lumezanu, C.; Cho, D.; Chen, H. Deep autoencoding gaussian mixture model for unsupervised anomaly detection. In Proceedings of the International Conference on Learning Representations, Vancouver, BC, Canada, 3 May–30 April 2018.
45. Su, Y.; Zhao, Y.; Niu, C.; Liu, R.; Sun, W.; Pei, D. Robust anomaly detection for multivariate time series through stochastic recurrent neural network. In Proceedings of the 25th ACM SIGKDD international conference on knowledge discovery & data mining, Anchorage, AK, USA, 4–8 August 2019; pp. 2828–2837.
46. Zhou, S.; Wang, T.; Yang, L.; He, Z.; Cao, S. A self-supervised fault detection for UAV based on unbalanced flight data representation learning and wavelet analysis. *Aerospace* **2023**, *10*, 250. [[CrossRef](#)]
47. Du, W.; Zhang, J.; Meng, G.; Zhang, H. Aero-engine fault detection with an LSTM auto-encoder combined with a self-attention mechanism. *Machines* **2024**, *12*, 879. [[CrossRef](#)]
48. Allka, X.; Ferrer-Cid, P.; Barcelo-Ordinas, J.M.; Garcia-Vidal, J. Pattern-based attention recurrent autoencoder for anomaly detection in air quality sensor networks. *IEEE Trans. Netw. Sci. Eng.* **2024**, *11*, 6372–6381. [[CrossRef](#)]
49. Yang, L.; Li, S.; Li, C.; Zhu, C.; Zhang, A.; Liang, G. Data-driven unsupervised anomaly detection and recovery of unmanned aerial vehicle flight data based on spatiotemporal correlation. *Sci. China Technol. Sci.* **2023**, *66*, 1304–1316. [[CrossRef](#)]
50. Reshef, D.N.; Reshef, Y.A.; Finucane, H.K.; Grossman, S.R.; McVean, G.; Turnbaugh, P.J.; Lander, E.S.; Mitzenmacher, M.; Sabeti, P.C. Detecting novel associations in large data sets. *Science* **2011**, *334*, 1518–1524. [[CrossRef](#)] [[PubMed](#)]
51. Gao, L.; Xu, C.; Wang, F.; Wu, J.; Su, H. Flight data outlier detection by constrained LSTM-autoencoder. *Wirel. Netw.* **2023**, *29*, 3051–3061. [[CrossRef](#)]
52. Zhong, J.; Zhang, Y.; Wang, J.; Luo, C.; Miao, Q. Unmanned aerial vehicle flight data anomaly detection and recovery prediction based on spatiotemporal correlation. *IEEE Trans. Reliab.* **2021**, *71*, 457–468. [[CrossRef](#)]
53. Yang, J.; Du, Y.; Qian, Y.; Li, F.; Zhu, Z.; Huang, H.; Song, Z. LSTM-Based Risk Identification for Civil Aircraft Tire Speed Limit Exceedance During Takeoff. *Int. J. Aeronaut. Space Sci.* **2026**, *27*, 3708–3720. [[CrossRef](#)]
54. Martinez, D.; Fernández, A.; Hernández, P.; Cristóbal, S.; Schwaiger, F.; Nunez, J.M.; Ruiz, J.M. Forecasting unstable approaches with boosting frameworks and lstm networks. In Proceedings of the 9th SESAR Innovation Days, Athens, Greece, 2–5 December 2019.
55. Chen, N.; Man, Y.; Li, J. Risk assessment method for civil aircraft approach and landing at high plateau based on QAR data. *J. Appl. Artif. Intell.* *Учредителю Cresta Press* **2024**, *1*, 151–168. [[CrossRef](#)]
56. Zhou, X.; Sun, J.; Zhao, J.; Shuang, F. Proactive Early Warning of Vortex Ring State in Coaxial UAVs: A Physics-Informed Multimodal ViT-LSTM Approach. *Sensors* **2026**, *26*, 3888. [[CrossRef](#)] [[PubMed](#)]
57. Wang, B.; Wang, Z.; Liu, L.; Liu, D.; Peng, X. Data-driven anomaly detection for UAV sensor data based on deep learning prediction model. In Proceedings of the 2019 Prognostics and System Health Management Conference (PHM-Paris), Paris, France, 2–5 May 2019; pp. 286–290.
58. Li, Y. A UAV sensor fault detection model based on dual-threshold method and LSTM. In Proceedings of the 2024 6th International Conference on Communications, Information System and Computer Engineering (CISCE), Guangzhou, China, 10–12 May 2024; pp. 1149–1152.
59. Guo, K.; Wang, N.; Liu, D.; Peng, X. Uncertainty-aware LSTM based dynamic flight fault detection for UAV actuator. *IEEE Trans. Instrum. Meas.* **2022**, *72*, 1–13. [[CrossRef](#)]
60. Azar, A.T.; Koubaa, A.; Ali Mohamed, N.; Ibrahim, H.A.; Ibrahim, Z.F.; Kazim, M.; Ammar, A.; Benjdira, B.; Khamis, A.M.; Hameed, I.A.; et al. Drone deep reinforcement learning: A review. *Electronics* **2021**, *10*, 999. [[CrossRef](#)]
61. Li, K.; Ni, W.; Dressler, F. LSTM-characterized deep reinforcement learning for continuous flight control and resource allocation in UAV-assisted sensor network. *IEEE Internet Things J.* **2021**, *9*, 4179–4189. [[CrossRef](#)]
62. Benyamen, H.; Chowdhury, M.; Keshmiri, S. Data-Driven Aircraft Modeling for Robust Reinforcement Learning Control Synthesis With Flight Test Validation. *J. Dyn. Syst. Meas. Control* **2024**, *146*, 061105. [[CrossRef](#)]

63. González-Baldizón, Y.; Fragoso-Mandujano, J.A.; Urbina-Brito, N.; Chandomí-Castellanos, E.; Bermúdez-Rodríguez, J.I.; Pérez-Pérez, E.J.; Guzmán-Rabasa, J.A. A Hybrid Algorithm for Fault Diagnosis in Nonlinear UAV Systems Using Conditional LSTM Autoencoders. *Algorithms* **2026**, *19*, 463. [\[CrossRef\]](#)
64. Yang, L.; Li, S.; Zhu, C.; Liu, J.; Zhang, A. Multi-source ensemble transfer learning-based unmanned aerial vehicle flight data anomaly detection with limited data: From simulation to reality. *Adv. Eng. Inform.* **2025**, *65*, 103255. [\[CrossRef\]](#)
65. Berndt, D.J.; Clifford, J. Using dynamic time warping to find patterns in time series. In Proceedings of the 3rd International Conference on Knowledge Discovery and Data Mining, Seattle WA, USA, 31 July–1 August 1994; pp. 359–370.
66. Deraz, A.A.; Badawy, O.; Elhosseini, M.A.; Mostafa, M.; Ali, H.A.; El-Desouky, A.I. Deep learning based on LSTM model for enhanced visual odometry navigation system. *Ain Shams Eng. J.* **2023**, *14*, 102050. [\[CrossRef\]](#)
67. Khosravi, A.; Nahavandi, S.; Creighton, D.; Atiya, A.F. Comprehensive review of neural network-based prediction intervals and new advances. *IEEE Trans. Neural Netw.* **2011**, *22*, 1341–1356. [\[CrossRef\]](#) [\[PubMed\]](#)
68. Gal, Y.; Ghahramani, Z. Dropout as a bayesian approximation: Representing model uncertainty in deep learning. In Proceedings of the International Conference on Machine Learning (PMLR), New York, NY, USA, 20–22 June 2016; pp. 1050–1059.
69. Raissi, M.; Perdikaris, P.; Karniadakis, G.E. Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations. *J. Comput. Phys.* **2019**, *378*, 686–707. [\[CrossRef\]](#)
70. Bach, S.; Binder, A.; Montavon, G.; Klauschen, F.; Müller, K.R.; Samek, W. On pixel-wise explanations for non-linear classifier decisions by layer-wise relevance propagation. *PLoS ONE* **2015**, *10*, e0130140. [\[CrossRef\]](#) [\[PubMed\]](#)
71. Lundberg, S.M.; Lee, S.I. A unified approach to interpreting model predictions. *Adv. Neural Inf. Process. Syst.* **2017**, *30*. Available online: <https://dl.acm.org/doi/10.5555/3295222.3295230> (accessed on 30 July 2026).
72. Chen, J.; Ran, X. Deep learning with edge computing: A review. *Proc. IEEE* **2019**, *107*, 1655–1674. [\[CrossRef\]](#)
73. Deng, L.; Li, G.; Han, S.; Shi, L.; Xie, Y. Model compression and hardware acceleration for neural networks: A comprehensive survey. *Proc. IEEE* **2020**, *108*, 485–532. [\[CrossRef\]](#)
74. Choi, H.C.; Deng, C.; Park, H.; Hwang, I. Stochastic conformal anomaly detection and resolution for air traffic control. *Transp. Res. Part C Emerg. Technol.* **2023**, *154*, 104259. [\[CrossRef\]](#)
75. Huang, H.; He, S.; Miao, F. CUQDS: Conformal uncertainty quantification under distribution shift for trajectory prediction. In Proceedings of the AAAI Conference on Artificial Intelligence, Philadelphia, PA, USA, 25 February–4 March 2025; Volume 39, pp. 17422–17430.
76. Zhou, Z.; Chong, X.; Chen, Z.; Zhou, J.; Zhang, J.; Guo, P. A Long Sequence Time-Series Forecasting Method for Early Warning of Long Landing Risks with QAR Flight Data. *Aerospace* **2025**, *12*, 744. [\[CrossRef\]](#)
77. Shankar, R.S.; Siotia, V.; Panicker, R.O. Drone Flight Dataset and Lightweight LSTM-Based Wind Estimation for Semi-Autonomous Quadcopter Control. *IEEE Access* **2025**, *13*, 203057–203076. [\[CrossRef\]](#)
78. Champion, M.; Ranganathan, P.; Faruque, S. UAV swarm communication and control architectures: A review. *J. Unmanned Veh. Syst.* **2018**, *7*, 93–106. [\[CrossRef\]](#)

Disclaimer/Publisher’s Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.