



Sung-Joo Lee

Dynamic Behaviour Classification in Multi-Domain
Operations Using ADS-B and Air Traffic Data

School of Aerospace, Transport and Manufacturing
Advanced Air Mobility Systems

MSc

Academic Year: 2024–2025

Supervisors: Prof. Antonios Tsourdos, Dr. Hae-In Lee and
Dr. Adolfo Perrusquia

Associate Supervisors: Dr. Andrew Hill and Dr. Andrew
Jenkins

August 2025



School of Aerospace, Transport and Manufacturing
Advanced Air Mobility Systems

MSc

Academic Year: 2024–2025

Sung-Joo Lee

Dynamic Behaviour Classification in Multi-Domain
Operations Using ADS-B and Air Traffic Data

Supervisors: Prof. Antonios Tsourdos, Dr. Hae-In Lee and
Dr. Adolfo Perrusquia

Associate Supervisors: Dr. Andrew Hill and Dr. Andrew
Jenkins

August 2025

This thesis is submitted in partial fulfilment of the
requirements for the degree of MSc.

© Cranfield University 2025. All rights reserved. No part of
this publication may be reproduced without the written per-
mission of the copyright owner.

Academic Integrity Declaration

I declare that:

- the thesis submitted has been written by me alone.
- the thesis submitted has not been previously submitted to this university or any other.
- that all content, including primary and/or secondary data, is true to the best of my knowledge.
- that all quotations and references have been duly acknowledged according to the requirements of academic research.

I understand that to knowingly submit work in violation of the above statement will be considered by examiners as academic misconduct.

Abstract

The increasing complexity of air traffic in multi-domain operations (MDO) demands real-time monitoring and classification of aircraft behaviour to support safety, efficiency, and situational awareness. Conventional approaches to conformance monitoring and behaviour classification are largely batch-oriented, relying on full-trajectory analysis and neglecting operational context, which limits their applicability during evolving flight operations. This dissertation proposes a dual-pipeline, AI-based framework that integrates unsupervised anomaly detection for conformance monitoring and supervised behaviour classification using Automatic Dependent Surveillance-Broadcast (ADS-B) and air traffic control (ATC) context. An LSTM-Autoencoder (LSTM-AE), incorporated with Notice to Airmen (NOTAMs) constraint, models nominal kinematics and flags deviations via reconstruction error, while a Bi-directional LSTM (Bi-LSTM) classifies behaviour at each time instant. To enable real-time inference, both models operate on overlapping sliding windows, providing early decisions while preserving short-term temporal dependencies. Experiments on ADS-B trajectories from the OpenSky Network show that the proposed framework achieves strong classification performance and the LSTM-AE reliably filters non-nominal patterns via elevated anomaly scores, by incorporating NOTAM-derived constraints improves operational relevance. These results demonstrate the feasibility of real-time, context-aware trajectory monitoring for MDO environments.

Keywords:

Advanced Air Mobility; Conformance Monitoring; Anomaly Detection; Trajectory Classification; Auto-Encoder; Long Short-Term Memory;

Acknowledgements

I would like to express my sincere gratitude to my academic supervisors, Prof. Antonios Tsourdos, Dr. Hae-In Lee, and Dr. Adolfo Perrusquia, for their invaluable guidance, support, and warm encouragement throughout the course of this research and dissertation. Their expertise, critical insights, and constructive feedback have been instrumental in shaping both the direction and the quality of this work. I am especially grateful for their mentorship, which has helped me grow not only as a researcher but also as a professional in the field of advanced air mobility systems.

I would like to thank my associate supervisors from Lockheed Martin UK Limited, Dr. Andrew Hill and Dr. Andrew Jenkins, for their insightful discussions and technical advice, which have greatly enriched this project.

I am also deeply grateful to Prof. Won-Sang Ra and Dr. Ick-Ho Whang for their infinite support in enabling my participation in this dual master's programme. Despite the distance, their thoughtful advice and critical insights have continuously encouraged the development of practical research outcomes and analytical insights.

I extend my heartfelt appreciation to my family—Myung-Won, Choo-Sung, and Sung-Hyeon—for their unwavering love, support, and encouragement throughout this academic journey. I would also like to thank my friends, Hyo-Won and Kwang-Hee, for their generous encouragement, support, and encouragement, which helped me adapt smoothly to studying abroad. Lastly, all glory and thanks to the Lord, my constant companion through every step of life.

Table of Contents

Academic Integrity Declaration	i
Abstract	ii
Acknowledgements	iii
Table of Contents	iv
List of Figures	vi
List of Tables	vii
List of Abbreviations	viii
1 Introduction	1
1.1 Project Background and Motivations	1
1.2 Aims and Objectives	2
1.3 Contributions	2
2 Literature Review	3
2.1 Conformance Monitoring	3
2.2 Behaviour Classification	6
2.3 Automatic Dependent Surveillance–Broadcast	8
2.4 Air Traffic Control Data	9
2.5 Challenges and Gaps	10
3 Problem Formulation	11
3.1 Assumptions and Scenarios	11
3.2 Key Performance Indicators	12
3.3 Learning-based Detection and Classification	13
3.4 Conclusion	13
4 Methodology	14
4.1 Data Preprocessing	14
4.2 Network Architecture	19
4.3 Network Training	22
4.4 Conclusion	23

5	Simulation	24
5.1	Conformance Monitoring	24
5.2	Behaviour Classification	26
5.3	Conclusion	32
6	Conclusion	33
6.1	Summary of Contributions	33
6.2	Key Findings and Limitations	33
6.3	Recommendations and Future Work	34
	References	35

List of Figures

4.1	Departure trajectory scenario and altitude	15
4.2	Arrival trajectory scenario and corresponding kinematic time series	16
4.3	Anomalous flight behaviour within the arrival phase of a trajectory	17
4.4	Distribution of behaviour classes	18
4.5	Sliding window input structure ($N = 1$)	19
4.6	Schematic of proposed network structure	20
4.7	LSTM-AE network structure for conformance monitoring	20
4.8	Bi-LSTM network structure for behaviour classification	21
4.9	Class distribution across training and test datasets	22
5.1	Conformance monitoring results: scenario views (top) and anomaly probabilities from GAE and LSTM-AE (bottom)	25
5.2	NOTAMs induced anomaly probability (LSTM-AE)	25
5.3	Behaviour classification performance on the training dataset; confusion matrix (left) and per-class precision, recall, F1-score, and accuracy (right)	27
5.4	Behaviour classification performance on the test dataset; confusion matrix (left) and per-class precision, recall, F1-score, and accuracy (right)	28
5.5	Behaviour classification results for the departure scenario	29
5.6	Behaviour classification results for the arrival scenario	30
5.7	Monte-Carlo class probabilities for Bi-LSTM across window size	31

List of Tables

2.1	A summary of conformance monitoring by operational mode	5
2.2	A summary of behaviour classification literature by operational mode . .	7
2.3	Examples of ADS–B datasets/providers and characteristics	8
2.4	Examples of ATC data types and their descriptions	9
4.1	Behaviour class definitions and conditions	15
4.2	Network hyperparameters used for anomaly detection (LSTM-AE) and behaviour classification (Bi-LSTM)	23

List of Abbreviations

AE	Autoencoder
AAM	Advanced Air Mobility
ADS-B	Automatic Dependent Surveillance–Broadcast
AIP	Aeronautical Information Publication
AI	Artificial Intelligence
ARTAS	ATM Surveillance Tracker and Server
ATC	Air Traffic Control
ATM	Air Traffic Management
Bi-LSTM	Bi-directional Long Short-Term Memory
CCLP	Compact Clustering via Label Propagation
CNN	Convolutional Neural Network
CVAE	Convolutional Variational Autoencoder
DAE	Discriminatory Autoencoder
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DL	Deep Learning
DTW	Dynamic Time Warping
FAA	Federal Aviation Administration
FOQA	Flight Operational Quality Assurance
GAE	Graph Autoencoder
GMM	Gaussian Mixture Model
GNSS	Global Navigation Satellite System
GNN	Graph Neural Network
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
ICAO	International Civil Aviation Organization
kNN	k-Nearest Neighbour
KPI	Key Performance Indicator
KSFO	San Francisco International Airport
LSTM	Long Short-Term Memory
LSTM-AE	LSTM Autoencoder
MAP	Maximum a Posteriori
MDO	Multi-Domain Operations
ML	Machine Learning
MMTS	Multimodal Model on Trajectory Sequence
MSE	Mean Squared Error
NM	Network Manager
NN	Neural Network
NOTAM	Notice to Airmen

SR-GNN	Spectral Residual Graph Neural Network
STGNN	Spatio-Temporal Graph Neural Network
SVM	Support Vector Machine
SVDD	Support Vector Data Description
TAACKIT	Track Annotation and Analytics with Continuous Knowledge Integration Tool
TFM	Traffic Flow Management
TP	True Positive
FP	False Positive
FN	False Negative
TN	True Negative
UTC	Coordinated Universal Time

Chapter 1

Introduction

1.1 Project Background and Motivations

As international military threats becomes more complex, large-scale, and frequent, the demand for multi-domain operations (MDO) has grown significantly. MDO integrate capabilities across the air, land, maritime, space, and cyber domains to achieve operational superiority in highly contested and dynamic environments [1]. These operations require automated, real-time situational awareness and decision-making tools capable of interpreting diverse, dynamic operational behaviours.

Automatic Dependent Surveillance-Broadcast (ADS-B) and air traffic data have discovered as essential information for real-time monitoring of aerial activities. These systems provide rich spatiotemporal information, such as position, velocity, and aircraft identity, enabling the tracking and analysis of aircraft behaviours. However, despite their widespread adoption, using these data for accurate and robust behaviour classification remains challenging due to factors such as unpredictable anomalous behaviour implied in multi-agent environments [2–4].

Recent research in conformance monitoring has predominantly focused on unsupervised learning approaches, including clustering techniques and deep learning architectures such as autoencoders and LSTMs [2, 3, 5, 6]. While effective in offline settings, these methods typically operate in batch mode and lack the responsiveness required for real-time anomaly detection. Moreover, few studies explicitly address dynamic temporal discrepancies across aircraft behaviours in multi-agent environments, a gap this research aims to address by incorporating autoencoder-based network [4, 7]. In contrast, most behaviour or trajectory classification studies perform analysis over entire flight sequences in a batch-processing methods. Such approaches overlook the fact that behavioural transitions in aircraft often occur over short, discrete intervals.

To develop enhanced situational awareness and support autonomous decision-making for MDO environments, this research focuses on dynamic conformance monitoring and behaviour classification using ADS-B and air traffic data. By developing the artificial intelligence (AI) approaches that capture the hidden patterns within multiple feature sequences will contribute to more resilient and intelligent air traffic management (ATM) systems and efficient MDOs.

1.2 Aims and Objectives

The aims and objectives of this dissertation are to develop an AI-based framework for real-time conformance monitoring and behaviour classification of aircraft using ADS-B and air traffic data, to enhance situational awareness and decision-making in MDO. To achieve these aims, the following objectives are defined:

- Review state-of-the-art approaches in conformance monitoring or anomaly detection, and behaviour or trajectory classification, focusing on the applications to ADS-B dataset.
- Investigate challenges and limitations of existing approaches that are applicable for the real-time monitoring in MDO environments.
- Design and implement a dual-pipeline framework; Long Short-Term Memory Autoencoder (LSTM-AE) for anomaly detection and Bi-directional LSTM (Bi-LSTM) network for behaviour classification of commercial flights.
- Integrate Notice to Airmen (NOTAMs) data into ADS-B datasets to simulate and benchmark anomaly detection.
- Optimise the hyperparameters of the network architecture and training procedure for improved performance and robustness.
- Evaluate the proposed framework in terms of performance indices (e.g., precision, recall, F1-score), compared with the existing methodologies.

1.3 Contributions

This dissertation presents several key contributions to the field of MDO, focusing on the real-time anomaly detection and classification of commercial aircraft behaviours using ADS-B and air traffic data. First, an AI-based framework integrating LSTM-AE for anomaly detection and Bi-LSTM network for behaviour classification of commercial flights using OpenSky ADS-B data. Second, this proposed framework achieves real-time detection and classification of flight behaviours, enabling enhanced situational awareness in multi-domain operational environments. Third, integration of ADS-B data with NOTAMs to simulate and evaluate anomalous scenarios is proposed, which provides a unique dataset for benchmarking. Finally, performance analysis on different hyperparameters is suggested to optimise the performance of the proposed framework, demonstrating its robustness and scalability in dynamic operational environments.

In the following chapters, Chapter 2 presents a literature review on anomaly detection, behaviour classification, and the dataset review on the ADS-B and air traffic data. Chapter 3 formulates the problem, outlining scenarios, assumptions, and key performance indicators (KPI). Chapter 4 details the methodology adopted for this project. It includes dataset pre-processing, network architecture, training procedure of the network, and ablation study for the network performance optimisation. Chapter 5 represents simulation results with the analysis on KPIs to the aforementioned simulation environment, including the real-time anomaly detection and behaviour classification. Finally, Chapter 6 concludes with key findings and future research directions.

Chapter 2

Literature Review

This chapter reviews existing studies on the real-time conformance monitoring and behaviour classification methodologies relevant to air traffic analysis. It also examines the publicly available datasets, such as ADS-B and air traffic data, that are commonly used in these applications. The chapter concludes with an analysis of the key challenges and research gaps identified in the current literature.

2.1 Conformance Monitoring

Conformance monitoring in air traffic control (ATC) refers to the detection of anomalous behaviours that deviate from a nominal flight path or the assigned flight plan, potentially compromising safety or operational efficiency [8]. Anomalous behaviours typically result from unforeseen influences—such as ATC clearance modifications, adverse weather conditions, or emergency rerouting—making labelled anomaly data rare or unavailable. This scarcity necessitates the use of unsupervised learning methods that infer deviations from baseline normal patterns.

As a conventional approach, distance-based and distributional methodologies have been widely used for conformal anomaly detection. One of the distance-based techniques involves clustering method, where the measured distance between data points and nominal clusters is used to quantify discrepancies. Among such methods, Density-Based Spatial Clustering of Applications with Noise (DBSCAN) groups data points based on density, identifying high-density regions as clusters and marking points in low-density areas as outliers [5]. Hierarchical DBSCAN (HDBSCAN) extends this approach by constructing a hierarchy of density-based clusters and selecting the one having the best stability, offering improved robustness to parameter adjustment and variable data densities [6]. As a distributional approach, Gaussian Mixture Models (GMM) assume that the data distribution can be modelled as a mixture of Gaussian components. These models estimate the probability that a given point belongs to each component and identify anomalies as those with low likelihoods [9]. However, these methods generally operate in batch mode, requiring access to the full dataset during the training procedure. As a result, they are not suitable for real-time conformance monitoring, where data must be processed sequentially as it arrives.

To ensure the real-time capability of anomaly detection in dynamic air traffic environments, several unsupervised approaches have been adapted to process incoming data

incrementally [3, 4, 10, 11]. These methods aim to identify behavioural deviations in aircraft operations as they occur, without requiring access to the entire complete flight trajectory. The underlying architecture of these methods mainly involves a scoring function that quantifies the level of deviation between the acquired data and a model trained on nominal patterns. Depending on how the model is trained and the nature of deviation it aims to capture, real-time unsupervised anomaly detection methods can be broadly categorised by the type of discrepancy they assess: reconstruction discrepancy, forecast discrepancy, and relational discrepancy.

Reconstruction discrepancy is designed to calculate the error between an observed input and its reconstructed output generated by a model trained on nominal patterns. The reconstructed output is expected to follow nominal behaviour observed in the training data, resulting in low reconstruction error. However, when an anomalous behaviour is encountered, the model fails to accurately reconstruct it, thereby producing a higher discrepancy. A common approach following this framework is the use of auto-encoder deep learning architecture, which compress and reconstruct input data to evaluate anomaly scores based on reconstruction loss [3, 12–15]. Discriminatory Auto-Encoder (DAE) trains separate decoders with the base structure of Long Short-Term Memory (LSTM)-based auto-encoder for distinct flight phases to improve anomaly detection across climbing, cruising, and descending on the ADS-B dataset [12, 13]. Convolutional Variational Auto-Encoder (CVAE) is designed to process multivariate time series data by leveraging convolutional layers for local temporal dependency and variational inference to model latent representations [3, 14]. By introducing a tunable regularisation parameter to balance reconstruction accuracy and generalisation, the CVAE enables fully unsupervised and real-time detection of anomalous flight behaviour based on reconstruction error thresholds. Furthermore, extending the VAE framework with Support Vector Data Description (SVDD) enhances both adaptability and detection accuracy by addressing a key limitation of VAEs. Their reliance on fixed reconstruction error thresholds, which limits sensitivity to diverse anomaly patterns [15].

Forecast discrepancy, on the other hand, is based on the residual error between the observed input and a one-step-ahead prediction. During anomalous behaviour, this error—measured in terms of absolute or mean-squared difference—tends to increase. A representative method in forecast discrepancy is based on the LSTM framework, which predicts future sequences and compares them to actual observations in real-time to compute anomaly scores. By applying a sliding window approach over overlapping temporal sequences, LSTM-based predictors have been used to detect simulated ADS-B spoofing attacks through elevated prediction residuals across various flight parameters [11]. An expansion of this framework incorporates an encoder-decoder structure to predict future sequences that reflect nominal behaviour patterns, thereby enhancing sensitivity to deviations [2, 16]. Another expansion applies bidirectional LSTM (Bi-LSTM) for trajectory prediction under ADS-B signal loss or spoofing conditions, where large residuals indicate anomalous behaviour [17]. The key distinction between reconstruction and forecast discrepancy lies in how the expected output is generated. The former reconstructs the input directly, while the latter generates future estimates based on historical context and compares them to the incoming observation.

In wider domains, Graph Neural Network (GNN)-based anomaly detection for time series data primarily operates on the principle that models learn what constitutes normal-

ity for a given dataset [18]. Reconstruction discrepancy trains a spatio-temporal GNN (STGNN) model to reproduce nominal inputs, while anomalies yield large reconstruction errors [19]. Variants differ in how they encode dependencies—such as attention, learned/sparse graphs—and in regularisation of low-dimensional codes and variational objectives, such as VGCRN or FuSAGNet [20, 21]. Forecast discrepancy instead trains a GNN forecaster for one-step-ahead prediction and scores anomalies by residuals between predictions and observations, such as GDN, GTA and CST-GL, with extensions of GST-Pro that anticipate future anomalies without ground-truth observations [22–25]. Hybrid designs, such as MTAD-GAT and DyGraphAD combine both discrepancies to improve robustness [26, 27]. However, GNNs are rarely adopted in aviation sections for anomaly detection due to the absence of aviation-specific graph formulation, and explainability/certification constraints favor simpler models [7].

Relational discrepancy assumes that anomalous behaviours manifest as deviations in the underlying relationships across multiple state variables, compared to those observed during nominal conditions. Unlike reconstruction or forecast-based approaches, which focus on individual variable patterns, relational discrepancy methods explicitly model the structural dependencies among variables over time. GNN is a powerful tool for relational discrepancy-based anomaly detection in aviation due to their ability to model complex, non-Euclidean dependencies between entities over time [7]. Unlike the conventional neural networks, GNNs perform message-passing over graph structures, enabling the detection of structural or interaction anomalies in evolving topologies [18]. Notably, Spectral Residuals GNN (SR-GNN) employs dynamic graph structures over ADS-B flight data to identify anomalous relational patterns in real-time without requiring labels [4].

Although unsupervised approaches are predominant in aviation anomaly detection due

Table 2.1: A summary of conformance monitoring by operational mode

Category	Representative Methods
Batch operation (Unsupervised)	Distance-based clustering: DBSCAN [5], HDBSCAN [6], Distribution-based: GMM [9]
Real-time operation (Unsupervised, Reconstruction error)	Autoencoder (AE) variants on ADS-B/FOQA [12, 13], Convolutional Variational AE (CVAE) [3, 14], VAE-SVDD hybrid [15]
Real-time operation (Unsupervised, Forecast error)	LSTM-based one-step-ahead predictors [11], LSTM encoder–decoder [2, 16], Bi-LSTM trajectory prediction under ADS-B loss [17]
Real-time operation (Unsupervised, Relational error)	SR-GNN with dynamic graph construction on ADS-B [4], GNN-based relational anomaly detection surveys [7, 18], Spatio-temporal GNN frameworks [22, 26]
Real-time operation (Semi-) Supervised	Semi-supervised M1+M2 deep generative model [28], CCLP for FOQA data [29], Supervised ADS-B anomaly detection with false data generator [30]

to limited labelled data, several supervised and semi-supervised methods are suggested. For example, semi-supervised frameworks such as M1+M2 deep generative model and Compact Clustering via Label Propagation (CCLP) leverage a small set of labelled flight data in combination with unlabelled Flight Operational Quality Assurance (FOQA) time-series data to detect anomalies during take-off and landing phases with high accuracy and low false-alarm rates [28, 29]. A fully supervised deep learning (DL) approach is developed using a false data generator to inject anomalies into ADS-B messages, and is trained on the resulting labelled dataset to detect these anomalies [30].

Conformance monitoring has progressed from batch clustering/mixture models to online, sequence-aware detectors, but deployment remains constrained. Classical DB-SCAN/HDBSCAN/GMM methods are batch-based operation and parameter-sensitive, limiting robustness under traffic/procedure drift [5, 6, 9]. Online DL reduces latency but has trade-off between window length and noise/delay, requiring continual threshold calibration [10]. Most detectors assume a stationary notion of normality; performance degrades as unexpected patterns evolve [3, 11]. Kinematics-only pipelines miss operational context, inflating false positives; ADS-B integrity issues—such as dropouts, mis-codes, and spoofing—further confound behaviour anomalies [2, 11, 16]. Relational methods are promising but hinge on graph-construction choices and can be compute-intensive online [4, 7, 18, 26]. Finally, many works lack latency-aware metrics, uncertainty calibration, and explanations, complicating thresholding and controller trust.

2.2 Behaviour Classification

Behaviour or trajectory classification in air traffic analysis focuses on identifying and categorising the dynamic flight patterns of aircraft of the given spatiotemporal trajectories to the pre-defined class candidates. This task is critical for enhancing situational awareness and supporting decision-making in both civil and multi-domain operational environments.

Early research in aircraft behaviour classification primarily focused on runway prediction using supervised learning approaches, applying various supervised machine learning (ML) and neural network (NN) classifiers, including convolutional neural networks (CNN), decision trees, support vector machines (SVM), and k-nearest neighbours (kNN), to categorise aircraft trajectories and predict intended landing runways [31]. More recently, the TAACKIT (Track Annotation and Analytics with Continuous Knowledge Integration Tool) framework has been introduced, leveraging unsupervised K-means clustering to generate preliminary behavioural labels—such as landing, take-off, and touch-and-go—which are subsequently validated and used to train supervised classifiers like SVM [32]. This approach significantly reduced manual annotation effort by over 90% while enhancing classification accuracy. Further studies have applied neural networks to classify aircraft motion dynamics into categories such as approach, landing, and taxi using ADS-B-derived kinematic features, enabling the differentiation of operational phases in terminal areas [33]. A CNN has also been trained on ADS-B dataset to distinguish operational patterns, including holding, manoeuvring, and nominal flight behaviour [34]. This approach focuses on identifying deviations from standard trajectories by leveraging spatial-temporal features. Additionally, machine learning methods, particularly SVM, have been applied to classify air traffic control conflict resolution manoeuvres, including horizontal actions (direct-to, path stretch), vertical adjustments (temporary level-off,

step altitude climb/descent, temporary altitude descent, extended altitude), and speed changes (change in cruise speed, temporary change in cruise speed, descent profile speed change) [35]. Flight phase classification has also been addressed using methods such as fuzzy logic and LSTM networks [36–38]. Despite these advancements, existing methodologies predominantly adopt batch-oriented processing. Such approaches are inherently limited, as they cannot effectively capture short-window behavioural transitions, which are critical for real-time air traffic monitoring and dynamic decision-making support.

One method for real-time behaviour classification involves employing a LSTM network to classify an aircraft’s landing direction [39]. This approach integrates dynamic time warping (DTW) and hierarchical clustering to effectively handle temporal misalignments in trajectory data, and is coupled with a real-time classification framework. A classification decision is made once the trajectory pattern probability exceeds a predefined threshold. Another study adopted a multimodal DL framework to classify typical aircraft behaviours, including direct flight, patrolling, and hovering before landing [40]. This framework, known as the Multimodal Model on Trajectory Sequence (MMTS) employs a transformer encoder to abstract multiscale features from the 1D geospatial coordinate sequences, considered as a first modality. In addition, these sequences are used to generate corresponding geographic map images, which serve as the second modality. While primarily evaluated offline, provides an architectural basis adaptable for real-time behaviour classification.

Existing behaviour classification studies demonstrate that classical ML/DL models can classify overall flight behaviours from ADS-B trajectories. However, most are developed in batch operation, providing limited capability for short-time decisions needed in real-time. Moreover, few works report latency-aware metrics or uncertainty calibration suited for real-time classification.

Table 2.2: A summary of behaviour classification literature by operational mode

Category	Representative Methods
Batch operation (Supervised)	Runway/intent classification with classical ML/DL [31], Motion-dynamics classification with DL [33], CNN for holding/manoeuvring patterns [34], SVM for conflict-resolution manoeuvre classification [35], Flight-phase identification (fuzzy logic, LSTM) [36–38]
Batch operation (Unsupervised)	TAACKIT: K-means for preliminary labels, followed by SVM [32]
Real-time operation (Supervised)	LSTM for landing-direction classification with DTW and hierarchical clustering [39]
Near real-time / Multimodal (Offline to Online)	MMTS; transformer encoder on coordinate sequences and image modality [40]

2.3 Automatic Dependent Surveillance–Broadcast

The ADS-B system is a pivotal advancement in new-generation air traffic surveillance, significantly enhancing situational awareness [41]. By continuously deriving state variables from the Global Navigation Satellite System (GNSS), ADS-B overcomes the limitations of radar in challenging environments like mountainous terrain or over water [16]. ADS-B delivers position and velocity data with high accuracy and frequent updates—typically every second—compared to traditional radar sampling intervals of 5–12 seconds [42]. This faster sampling rate and GPS-driven precision offer a more detailed and timely view of airspace dynamics, improving both safety and operational efficiency.

The ADS-B system comprises three elements: ADS-B Out, ADS-B In, and ground stations [43]. ADS-B Out is the onboard transmitter that periodically broadcasts an aircraft’s state—such as GNSS-derived position, altitude, ground speed, heading—along with unique aircraft identifiers such as the callsign and International Civil Aviation Organization (ICAO) 24-bit address. ADS-B In is an onboard receiver that allows cockpit displays to receive state broadcasts from other aircraft via ADS-B Out, as well as ground-based surveillance and weather data where available. Ground stations receive ADS-B Out transmissions and feed them to surveillance processing systems used by air traffic control.

Majority of current surveillance systems adopt ADS-B as a primary data source to enhance operational awareness and extend coverage. For example, EUROCONTROL’s ATM Surveillance Tracker and Server (ARTAS) fuses ADS-B with radar and multilateration inputs across Europe to maintain unified and high-accuracy airspace situation [44]. Meanwhile, Aireon’s space-based ADS-B network utilises satellite-mounted receivers to provide global, real-time aircraft tracking, including in remote and oceanic areas beyond ground infrastructure [45]. Additionally, the capstone program of Federal Aviation Administration (FAA) pioneered ADS-B deployment in rural and mountainous regions, combining onboard avionics and ground-based receivers to support real-time tracking where radar coverage is limited [46].

Table 2.3: Examples of ADS–B datasets/providers and characteristics

Dataset / Provider	Data Contents	Typical Cadence
OpenSky Network (Historical DB, REST/Impala)	Raw 1090ES messages; state vectors (lat/lon, alt, vel); Mode–S EHS fields	State vectors: ~0.2 Hz, raw messages: ~2 Hz per aircraft
ADS-B Exchange (API/websocket/archive)	Aggregated state vectors; raw message/track feeds	~1 Hz
Aireon (space-based ADS-B)	Fused space-based state vectors	~1 Hz
FlightAware Firehose (aggregated feeds)	Aggregated tracks/state vectors; enrichment (aircraft metadata)	~1 Hz (tiered service)
FR24 Data Services	Aggregated tracks/state vectors	~0.2 – 1 Hz

The OpenSky Network is selected as the primary ADS-B data source because of its comprehensive coverage, research-oriented access, and historical archive compared to the other providers listed in Tab. 2.3. OpenSky provides not only the access to the live state vectors using a free REST API, but also full historical access via an SQL interface upon request and approval. In addition, OpenSky’s dataset have been extensively adopted in research, which ensures reproducibility and methodology consistency.

2.4 Air Traffic Control Data

Air Traffic Control (ATC) data comprises operational information exchanged between airspace users, controllers, and network managers that describes intended behaviours, clearances, constraints, and network status. Complementary to ADS-B’s observed kine-

Table 2.4: Examples of ATC data types and their descriptions

ATC Data Type	Description
Flight Plan (FPL)	FPL specifies the planned trajectory to be followed under ATS unit, including departure and arrival aerodromes, route (airways/waypoints), requested flight level and speed, time estimates, SSR code, and equipment/capabilities. Associated messages (CHG/DLA/CNL) and route amendments reflect changes to the intended operation, providing a baseline for coordination and conformance monitoring [47].
Controller–Pilot Data Link Communications (CPDLC)	Digitally conveys ATC clearances (e.g., climb/descent, directs, speed/heading), route changes, and handovers between controller and pilot. CPDLC improves communication efficiency, reduces voice congestion, and supports automation [48].
Flow/Network Management (TFM/NM)	Regulations, slots and various traffic management measures (e.g., CTOT/TTOT, level caps, miles-in-trail) published by FAA Traffic Flow Management (TFM) or EUROCONTROL Network Manager (NM) provide demand–capacity context that shapes aircraft trajectories [49, 50].
Aeronautical Information (NOTAMs/AIP)	Operationally significant, time-bounded constraints—such as runway/taxiway closures, activation of temporary or restricted airspace, navaid outages, and temporal changes to procedures—promulgated through Notice To Airmen (NOTAMs) for immediate notification, or Aeronautical Information Publication (AIP) for information of lasting character [51, 52].
Surveillance/Track Services	Fused ATC tracks from multiple surveillance sources—such as radar, multilateration, ADS-B—provide quality-controlled identifiers and track continuity to support robust aircraft data association [44].

matics, ATC data captures intent and constraints, which are critical for conformance monitoring and behaviour classification in real-time. The examples of ATC data and their descriptions are summarized in Tab. 2.4.

Unlike ADS-B, ATC data are heterogeneous in format—such as free-text NOTAMs, structured FPL messages, CPDLC uplinks, and SWIM/NM B2B services—and vary in timeliness and reliability. Therefore, practical fusion requires time alignment, modeling and normalisation—for example, mapping FPL route/constraint fields and NOTAM polygons to geo-temporal rules that can be checked against sliding-window ADS-B states [50, 53]. Emerging information-sharing frameworks such as ICAO GOLD and SWIM/NM B2B interfaces improve standardisation and access, but residual inconsistencies and delivery delays should be accounted for real-time conformance checks [48, 50–52, 54].

In this project, NOTAMs is selected as the complementary ATC dataset because they provide operationally significant, time-bounded constraints—e.g., runway/taxiway closure, airspace activation, temporal procedure change—that ADS-B kinematics alone cannot capture, and that are explicitly intended for real-time operational use. In contrast, FPLs only reflect pre-filed intentions, but do not incorporate subsequent tactical, real-time constraints. CPDLC conveys controller clearances, but its limited availability and latency significantly affects for time-critical communications. TFM/NM data capture broad, system-wide restrictions, but are macro in nature and not tailored to constraints affecting individual flights.

2.5 Challenges and Gaps

Through the surveyed literatures, the challenges and gaps for the conformance monitoring and behaviour classification are followed. Classical density or distribution models to detect anomalous behaviours are predominantly trained in batch and periodically retrain patterns, which limits low-latency deployment in dynamic airspace. ATC-kinematics fusion is non-trivial, where ATC constraints are heterogeneous and time-bounded. Thus, robust fusion is required converting them into geospatial rules and aligning with ADS-B state vectors.

In summary, existing approaches to behaviour classification are limited by full-trajectory dependence, lack of calibrated confidence, and weak robustness to anomalies or operational constraints. Most works do not integrate ATC context such as NOTAMs, leading to false alerts when deviations are compliant with temporary restrictions. Furthermore, real-time requirements—e.g., low decision latency and reliability under data scarcity—are rarely addressed. These gaps motivate the development of a dual-pipeline framework for real-time anomaly detection and behaviour classification, enriched with operational context and confidence-aware outputs.

Chapter 3

Problem Formulation

This chapter formalises the problem under study by stating the assumptions and scenarios, defining the key performance indicators (KPIs) for evaluation, and describing the learning-based detection and classification method.

3.1 Assumptions and Scenarios

For the simulation, some assumptions are considered for commercial and cargo flights operating in the terminal area of San Francisco International Airport (KSFO). As data-related assumptions, the state vectors—such as geospatial coordinates, barometric altitude, velocity, ground speed, track and heading—are correctly decoded and time-stamped in UTC. Messages associated with the same ICAO 24-bit address correspond to a single aircraft and trajectory, and duplicate or corrupted messages are removed in preprocessing.

In operational or air traffic assumptions, the ADS-B Out for all flights is transmitted in sampling rate of 1 Hz. The selected flights operate under nominal terminal-area procedures for KSFO. Approaches or departures follow the standard routes and runway usage patterns characteristic of the study period, without emergency procedures or abnormal ATC interventions unless explicitly analysed. As modelling assumptions, flight behaviour labels are derived solely from kinematic ADS-B features. Missing ADS-B points are addressed with interpolation over short outages so as not to materially distort the original trajectory. Multi-class evaluation metrics (KPIs) are computed in a one-vs-rest manner, where for each behaviour class of interest, that class is treated as positive and all others as negative.

The base scenario for nominal operations comprises historical ADS-B trajectories of commercial aircraft arriving to or departure from KSFO, within a terminal area window of 25 NM radius from the aerodrome reference point and below 50,000 ft barometric altitude. The trajectories analysed in this study correspond to 12 March 2018, with all timestamps expressed in Coordinated Universal Time (UTC). As a representative anomalous behaviour, a go-around is considered, defined as a sustained climb initiated during the short-final segment of an arrival.

3.2 Key Performance Indicators

To quantitatively evaluate the performance of the proposed algorithms, widely used KPIs for classification problems are considered. Since a classification task results predictions in a discrete solution space, these KPIs assess the extent to which the predictions conform the ground truth. In this project, four commonly used measures are employed: accuracy, recall, precision, and F1-score.

From a classification perspective, four fundamental terms are defined: true positive (TP), false positive (FP), false negative (FN), and true negative (TN). For a given target class C , treating all other classes as “not C ”,

$$TP_C = \Pr(\hat{y} = C \wedge y = C), \quad (3.1)$$

$$FP_C = \Pr(\hat{y} = C \wedge y \neq C), \quad (3.2)$$

$$FN_C = \Pr(\hat{y} \neq C \wedge y = C), \quad (3.3)$$

$$TN_C = \Pr(\hat{y} \neq C \wedge y \neq C), \quad (3.4)$$

where $\Pr(\cdot)$ denotes the probability of the specified event in the population. In practice, it is estimated empirically as the relative frequency in the dataset. TP_C denotes instances with the predicted class $\hat{y}$ and its ground truth $y = C$ are equivalent as class C . FP_C represents a type I error, where it predicts as true class $\hat{y} = C$ while the ground truth is the others $y \neq C$. FN_C refers to type II error, where the prediction $\hat{y} \neq C$ mismatches to the class $y = C$. TN_C represents the correct rejection which the prediction and ground truth are all not C .

Based on these definitions, the KPIs for a multi-class classification problem can be expressed as

$$\text{Acc}(C) = \frac{TP_C + TN_C}{TP_C + TN_C + FP_C + FN_C}, \quad (3.5)$$

$$\text{Prec}(C) = \frac{TP_C}{TP_C + FP_C} = \Pr(y = C \mid \hat{y} = C), \quad (3.6)$$

$$\text{Rec}(C) = \frac{TP_C}{TP_C + FN_C} = \Pr(\hat{y} = C \mid y = C), \quad (3.7)$$

$$F_1(C) = \frac{2 \text{Prec}(C) \text{Rec}(C)}{\text{Prec}(C) + \text{Rec}(C)}, \quad (3.8)$$

where

- $\text{Acc}(C)$ – Accuracy; the proportion of correctly classified instances (both positive and negative) for class $y = C$
- $\text{Prec}(C)$ – Precision; the proportion of predicted class $\hat{y} = C$ instances that are truly class C
- $\text{Rec}(C)$ – Recall; the proportion of actual class $y = C$ instances correctly identified
- $F_1(C)$ – F1-score; the harmonic mean of precision and recall

Higher values, closer to 1, indicate better classification performance for each KPI.

3.3 Learning-based Detection and Classification

Let a single trajectory from the ADS-B dataset be represented as

$$\mathcal{D} = \{(\mathbf{x}_t, y_t)\}_{t=1}^N, \quad (3.9)$$

where $\mathbf{x}_t \in \mathbb{R}^D$ denotes the D -dimensional state vector, and $y_t \in \mathcal{C}$ refers to the ground-truth behaviour class label at time instant t . In this work, $\mathbf{x}_t$ contains the kinematic state variables from the OpenSky Network ADS-B data, including geospatial coordinates, barometric altitude, velocity, heading, and vertical speed.

The anomaly detection problem can be formulated as estimating the reconstruction discrepancy between the observed state vector $\mathbf{x}_t$ and its reconstructed vector $\hat{\mathbf{x}}_t$. The reconstruction error can be defined as the mean squared error (MSE) as

$$\mathcal{L}_t = \frac{1}{D} \|\mathbf{x}_t - \hat{\mathbf{x}}_t\|_F^2, \quad (3.10)$$

where $(\cdot)_F$ denotes a Frobenius norm between the state vector and its reconstruction. A decision as considering the $\mathbf{x}_t$ in an anomaly can be made when the anomaly score $\mathcal{L}_t$ exceeds a threshold that deviates from the score distribution. The anomaly score can be defined as

$$p_A(t) = 1 - \exp(-\alpha \mathcal{L}_t), \quad \alpha > 0, \quad (3.11)$$

where α is the design parameter for the anomaly score.

The behaviour classification problem can be formulated as estimating the posterior class probabilities $p(y_t = C | \mathbf{x}_t)$ for each class $C \in \mathcal{C}$. Under Bayes decision theory, the class estimate can be derived in terms of maximum a posteriori (MAP) estimation as

$$\hat{y}_t = \arg \max_{C \in \mathcal{C}} p(y_t = C | \mathbf{x}_t). \quad (3.12)$$

Given a trained model f_θ , the model produces a vector of logits

$$\mathbf{z}_t = f_\theta(\mathbf{x}_t) \in \mathbb{R}, \quad (3.13)$$

which are converted to a probability distribution over classes via the softmax function

$$p(y_t = C | \mathbf{x}_t) = \frac{\exp(z_{t,C})}{\sum_C \exp(z_{t,C})}, \quad (3.14)$$

where $z_{t,C}$ refers to the logit corresponding to class C as a component of $\mathbf{z}_t$.

3.4 Conclusion

This chapter formulates the behaviour classification problem for ADS-B trajectory data under defined assumptions and operational scenarios. The performance of classification algorithms is analysed using multi-class KPIs—precision, recall, and F1-score—derived from the underlying definitions of true/false positives and negatives. The learning-based classification is formulated as estimation problem, where class is predicted through MAP estimation and computed from model logits through the softmax function. These formulations provide the analytical basis for the model architecture, training, and evaluation strategies described in the following chapter.

Chapter 4

Methodology

This chapter presents the end-to-end methodology for developing a real-time conformance monitoring and behaviour classification framework using ADS-B data. It covers the preprocessing of raw flight trajectories from the OpenSky Network, the design of sequential neural network architectures, and the application of a sliding window mechanism for capturing temporal dependencies. The training procedure and an ablation study on key parameter choices are also discussed.

4.1 Data Preprocessing

As a preliminary step for training the network model, preprocessing the raw OpenSky ADS-B dataset is required. In this project, the traffic library is used to access historical OpenSky ADS-B data [55]. Due to the transmission interference between aircraft, internal errors such as erroneous measurements and irregular sampling rates are observed [56, 57]. To mitigate these issues, outliers that deviate significantly from the nominal trajectory are removed, and the data is downsampled to 0.1 Hz.

The kinematic state vector $\mathbf{x}_t$ at time instant t is defined as

$$\mathbf{x}_t \triangleq [t \quad d_t \quad h_t \quad \dot{h}_t \quad v_t \quad \psi_t \quad \dot{\psi}_t]^T, \quad (4.1)$$

where d_t represents the haversine distance between the aircraft and the KSFO airport at time instant t , computed as

$$d_t = 2R \cdot \sin^{-1} \left(\sqrt{\sin^2 \left(\frac{\phi_t - \phi_{\text{KSFO}}}{2} \right) + \cos \phi_t \cdot \cos \phi_{\text{KSFO}} \cdot \sin^2 \left(\frac{\lambda_t - \lambda_{\text{KSFO}}}{2} \right)} \right), \quad (4.2)$$

where R is the radius of the Earth, ϕ_t and λ_t denote the latitude and longitude of the aircraft, and ϕ_{KSFO} and λ_{KSFO} denote the latitude and longitude of KSFO, respectively. h_t and $\dot{h}_t$ represent the barometric altitude and vertical rate, v_t refers to the ground speed, and ψ_t is the heading angle. $\dot{\psi}_t$ is an appended state variable representing the heading rate, computed as the difference between the current and previous heading angles, normalised by the sampling time.

Additional identifiers included in the OpenSky ADS-B dataset are the ICAO 24-bit address—a unique transponder ID assigned to each aircraft—and the callsign, which represents the broadcast flight identifier. For this analysis, a unique ID is assigned to each

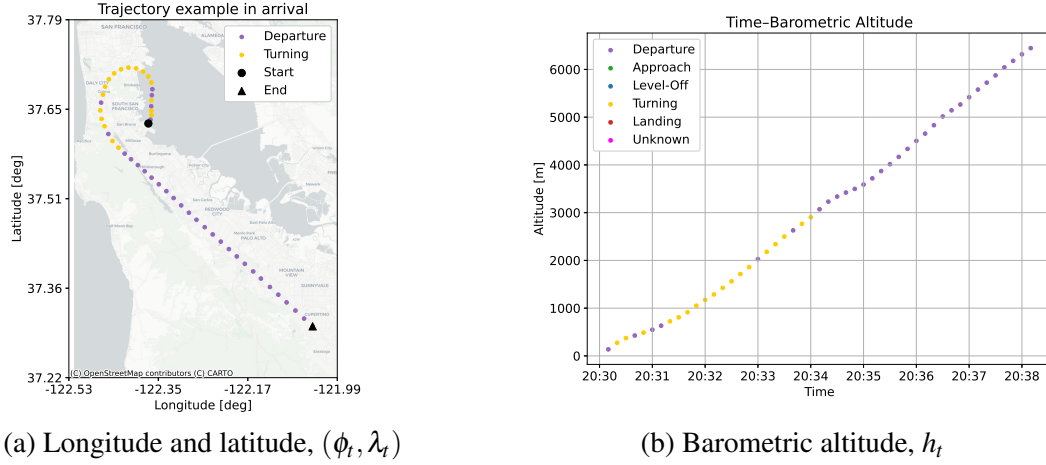


Figure 4.1: Departure trajectory scenario and altitude

trajectory which the callsign follows the ICAO telephony designator of Type C flight pattern, with the separation of at least 30 minutes to avoid overlap [58]. The flight trajectories of interest, whether arrival or departure to the KSFO airport, are filtered using the geospatial and altitude constraints as

$$d_t < 25 \text{ NM}, \quad h_t < 50,000 \text{ ft}, \quad (4.3)$$

The behaviour class y_t in Sec. 3.3 is defined according to the conditions described in Tab. 4.1, with illustrative examples depicted from Fig. 4.1 to Fig. 4.3. According to Fig. 4.1, the departure exhibits an aircraft ascending from KSFO, initially heading north before executing a counterclockwise turn to transition into an en-route phase toward the southeast. In contrast, the arrival trajectory approaches KSFO from the northwest, displaying a clear descent profile as it aligns with the runway.

Each segment of the trajectory can be distinguished through the temporal evaluation of key kinematic features in the state vector $\mathbf{x}_t$. As shown in Fig. 4.2d, the *Approach* phase is characterised by a negative vertical rate $\dot{h}_t$ below -3 m/s , indicating a steady descent while the heading rate remains near zero. When both the heading rate $\dot{\psi}_t$ and the vertical rate $\dot{h}_t$ remain within their respective thresholds, the aircraft is considered to be in the *Level-Off* phase, reflecting stable flight at a constant altitude in arrival. Notably, the heading rate $\dot{\psi}_t$

Table 4.1: Behaviour class definitions and conditions

Class ID	Label	Condition
C_1	Departure	$ \dot{\psi}_t < 0.3 \text{ deg/s}, \dot{h}_t > 3 \text{ m/s}$
C_2	Approach	$ \dot{\psi}_t < 0.3 \text{ deg/s}, \dot{h}_t < -3 \text{ m/s}$
C_3	Level-Off	$ \dot{\psi}_t < 0.3 \text{ deg/s}, \dot{h}_t < 3 \text{ m/s}$
C_4	Turning	$ \dot{\psi}_t > 0.3 \text{ deg/s}$
C_5	Landing	$h_t < 150 \text{ m}, v_t < 80 \text{ m/s}$
C_X	Unknown	Otherwise

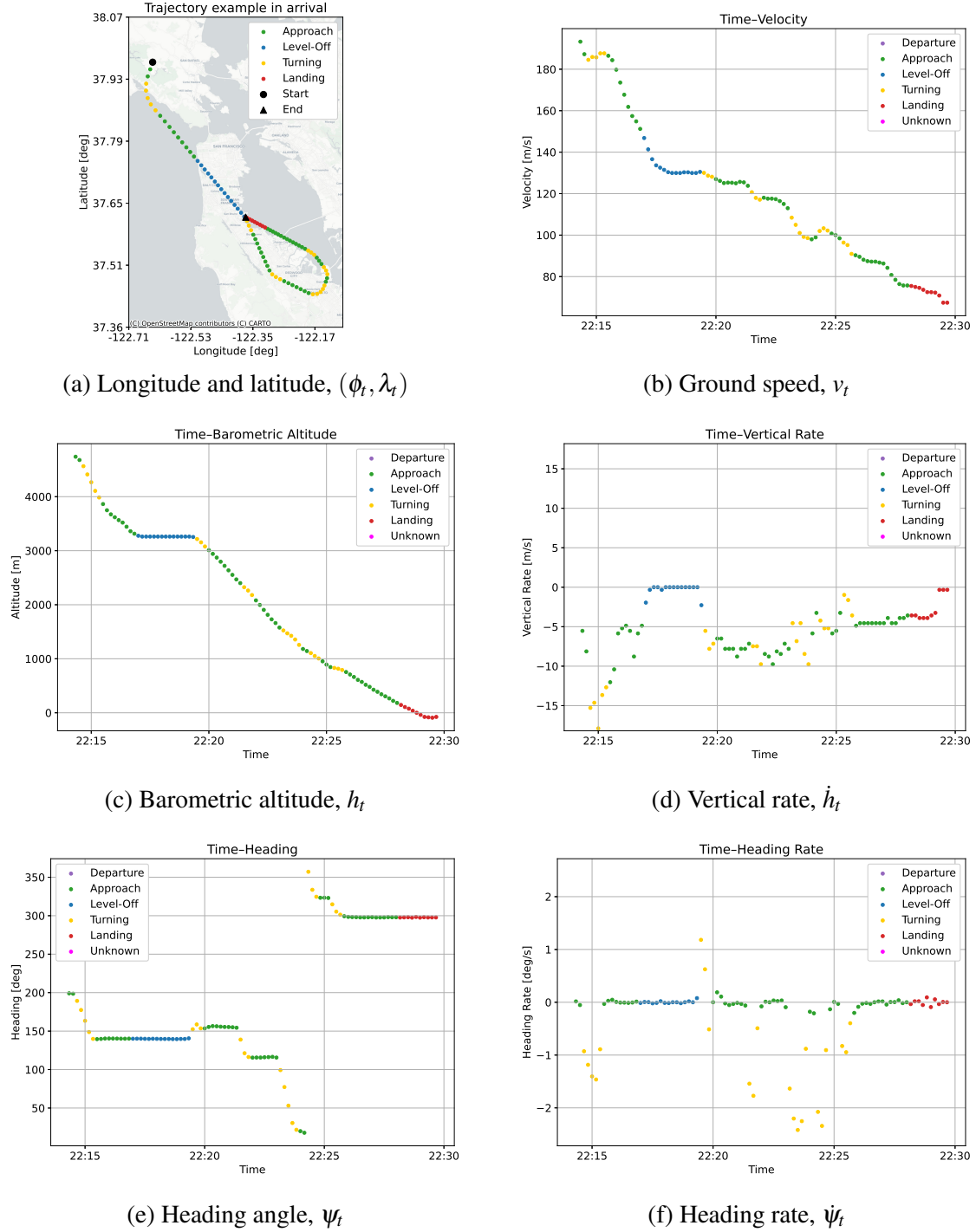


Figure 4.2: Arrival trajectory scenario and corresponding kinematic time series

is particularly informative for identifying *Turning* behaviours. As visualised in Fig. 4.2f, instances where $\dot{\psi}_t$ exceeds the threshold of 0.3 deg/s clearly indicate turning manoeuvres, especially prominent in curved segments during the approach. Finally, the *Landing* phase is observed in the terminal portion of the trajectory, where $h_t < 150$ m and $v_t < 80$ m/s, corresponding to the final descent and touchdown.

The anomalous behaviours are defined as class C_X , *Unknown*, when patterns are ob-

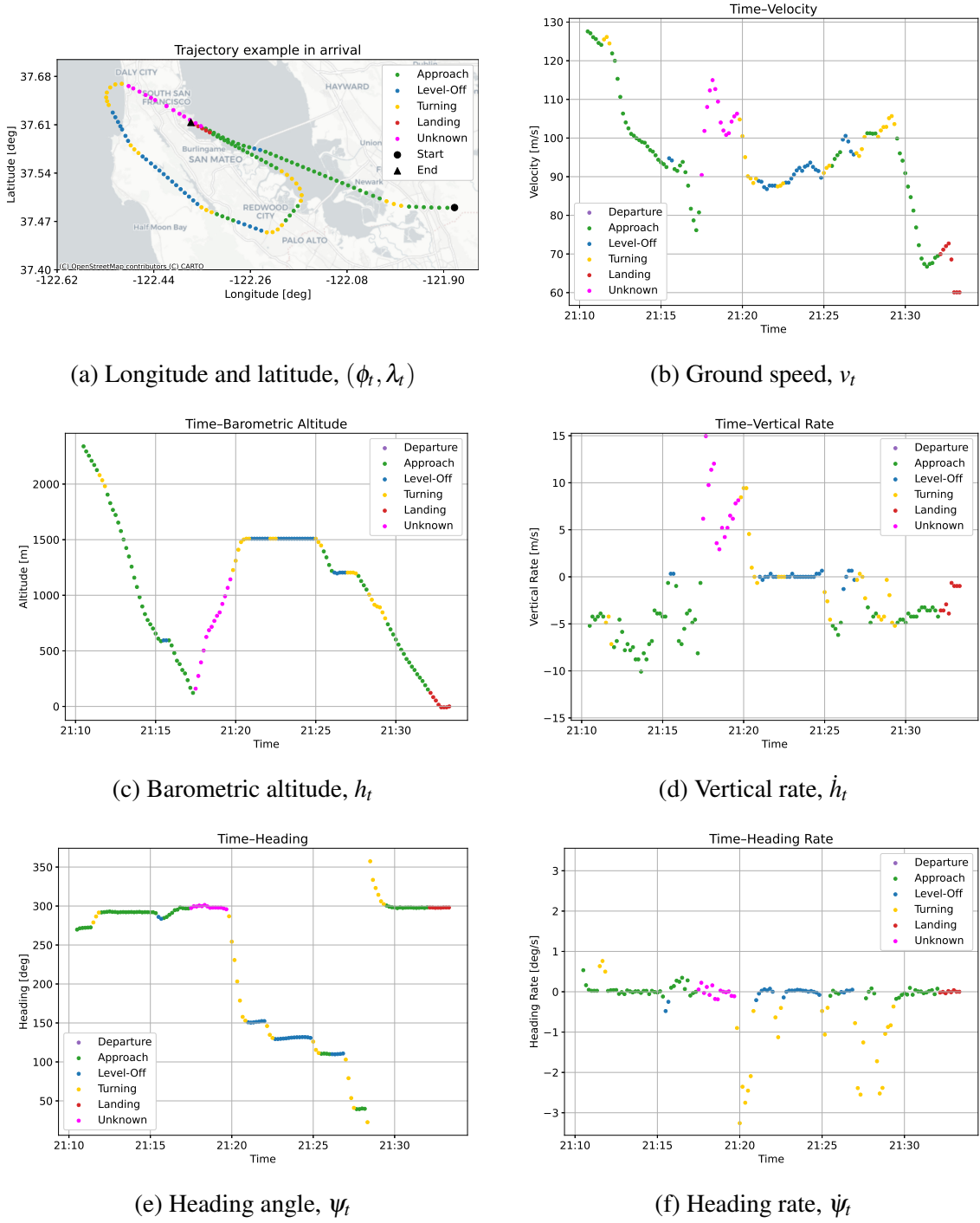


Figure 4.3: Anomalous flight behaviour within the arrival phase of a trajectory

served that deviate from the defined conditions of the nominal behaviour classes from C_1 to C_5 . Fig. 4.3 illustrates such an instance, where anomalous behaviour occurs during the arrival, commonly known as a go-around. Between 21:17 and 21:20, the aircraft unexpectedly increases its altitude, with $\dot{h}_t$ exceeding 3 m/s. This upward motion deviates from the expected descent pattern during a standard arrival, thereby triggering an *Unknown*.

The total number of trajectories considered is 365, with the distribution of class labels

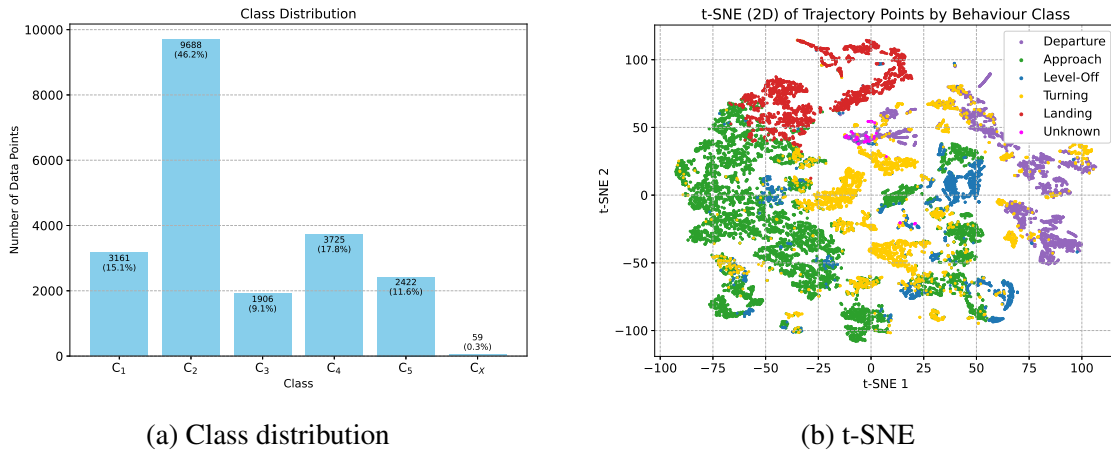


Figure 4.4: Distribution of behaviour classes

across all time instants illustrated in Fig. 4.4a. The majority of data points are associated with the *Approach* (C_2), accounting for approximately 46.2% of the dataset. This class imbalance implies that the classifier may develop a bias toward *Approach*, potentially leading to false alarms or misclassifications when distinguishing it from other classes with similar descent characteristics, such as *Landing* or *Level-Off*.

Fig. 4.4b depicts the two-dimensional t-SNE projection of the state vector $\mathbf{x}_t$, providing a qualitative perspective on the separability of behaviour classes [59]. Distinct clusters are observed for *Turning* and *Landing*, reflecting the strong discriminative feature of the heading rate $\dot{\psi}_t$ and altitude, velocity conditions. In contrast, *Approach* points are broadly distributed and overlap with *Turning*, as well as with *Level-Off* and *Landing*, highlighting the inherent ambiguity between steady descent and near-level segments. *Departure* trajectories are more distinct, generally separated from *Approach* due to their positive vertical rate. Meanwhile, the *Unknown* class does not form a coherent cluster, but scattered around decision boundaries, consistent with irregular behaviours such as go-arounds or noisy measurements.

4.2 Network Architecture

As reviewed in Sec. 2.2, most behaviour classification approaches in the literature rely on batch processing. To ensure the real-timeliness of the system, a sliding window-based approach is considered. Within a fixed-length window, the state vector is updated sequentially as a queue at every time instant. During training, the network learns temporal dependencies centred around each time instant, which ensures robustness compared to single-frame classification. Accordingly, the classification (3.12) can be reformulated as

$$\hat{y}_t = \arg \max_{C \in \mathcal{C}} p(y_t = C | \mathbf{x}_{t-N:t+N}), \quad (4.4)$$

where N denotes the half-length of the sliding window, resulting in a total window length of $2N + 1$. However, this structure introduces a classification latency proportional to the window size, as the prediction at time instant t can only be made after observing additional frames up to $t + N$.

An operation example of the sliding window with $N = 1$ is illustrated in Fig. 4.5. At time instants where the sliding window cannot be centred—such as at the beginning, t_1 or end of a trajectory, t_{f+2} —the class label is marked as N/A (i.e., not assigned due to insufficient temporal context). At time instant t_2 , the window is constructed as $[0, \mathbf{x}_1, \mathbf{x}_2]$, enabling the network to predict the class based on the centre state vector $\mathbf{x}_1$, resulting in label y_1 with a latency of one time step. This process continues sequentially until time instant t_f , which corresponds to the final valid window centred at $\mathbf{x}_{f-1}$, producing class label y_{f-1} . At time instant t_{f+1} , the prediction for class y_f is made using the final available window, even though the trajectory data ends at t_f . At t_{f+2} , classification becomes no longer feasible as the window cannot be centred on a valid state vector.

This project proposes a two-stage pipeline consisting of conformance monitoring and

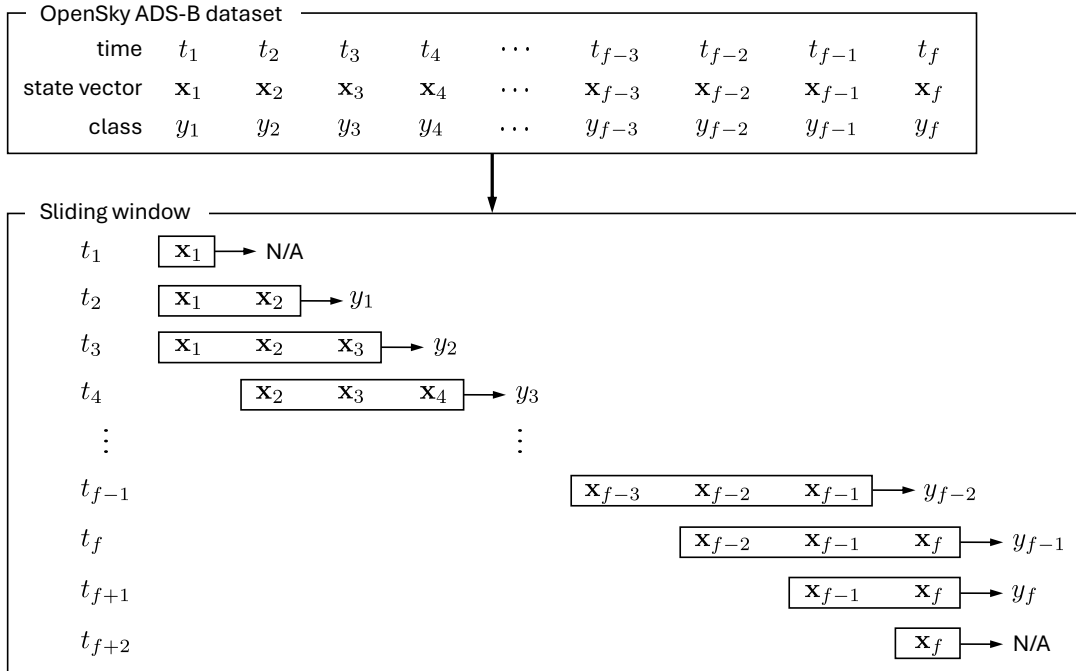


Figure 4.5: Sliding window input structure ($N = 1$)

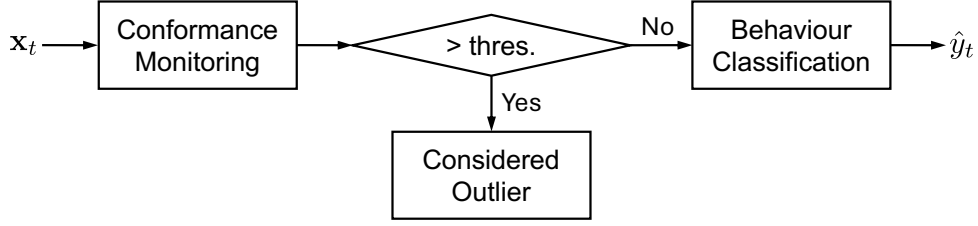


Figure 4.6: Schematic of proposed network structure

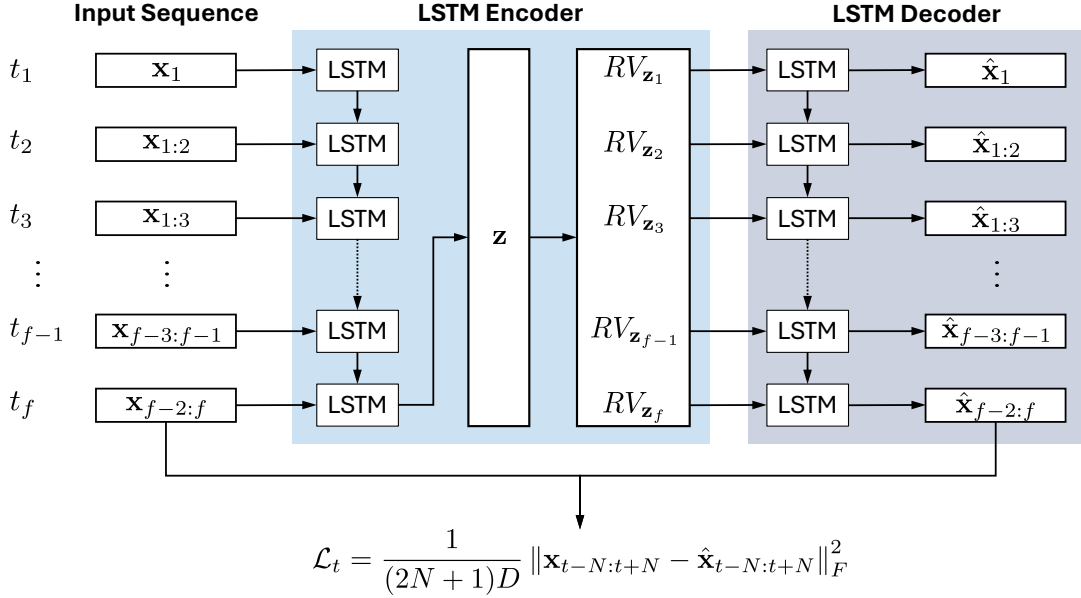


Figure 4.7: LSTM-AE network structure for conformance monitoring

behaviour classification, illustrated in Fig. 4.6. At each time instant t , the input state vector $\mathbf{x}_t$ is first processed by the conformance monitoring module to compute an anomaly score. If the score exceeds a predefined threshold, the trajectory is flagged as an outlier. Otherwise, the input proceeds to the behaviour classification module, which outputs the estimated behaviour label $\hat{y}_t$.

To enable real-time detection of anomalous flight behaviours and sequential classification of behaviour patterns, two neural network models are proposed in this study, a LSTM Autoencoder (LSTM-AE) for conformance monitoring and a bidirectional LSTM (Bi-LSTM) for behaviour classification. Both architectures are designed to handle time-series data efficiently, capturing temporal dependencies inherent in ADS-B dataset.

The LSTM-AE architecture, illustrated in Fig. 4.7, consists of an encoder-decoder framework. The encoder compresses the input sequence of state vectors within a window $\mathbf{x}_{t-N:t+N}$ into a latent representation $\mathbf{z}$, while the decoder reconstructs the sequence from this embedding to produce $\hat{\mathbf{x}}_{t-N:t+N}$. To initialise the decoder with sequential inputs, the latent vector $\mathbf{z}$ is duplicated across all time steps through a repeat vector mechanism ($RV_{\mathbf{z}_1}, \dots, RV_{\mathbf{z}_f}$). Each repeated copy of $\mathbf{z}$ serves as the input at a decoder time step, enabling the recurrent dynamics of the LSTM to unfold the reconstruction of the full sequence. The reconstruction error is quantified as $\mathcal{L}_t$, and if it exceeds threshold, the input sequence is considered as anomaly.

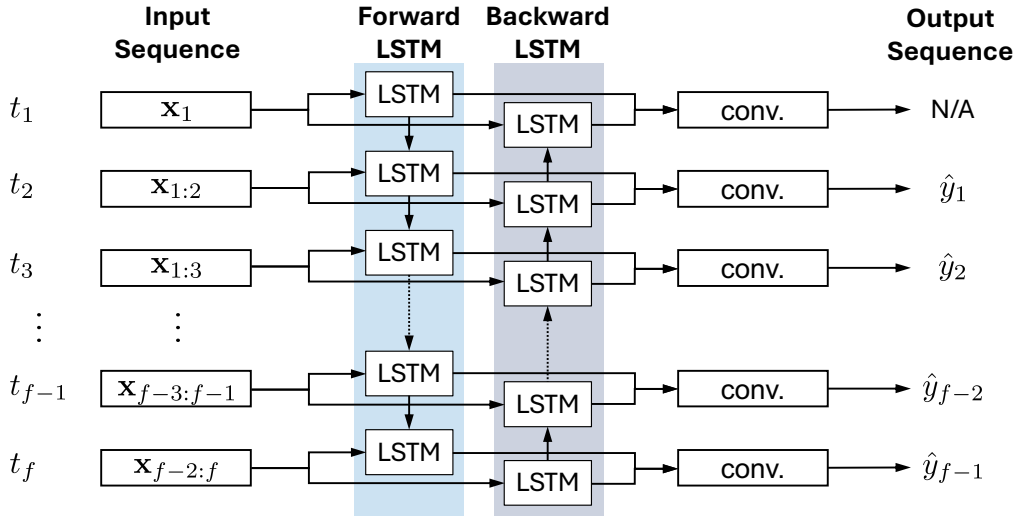


Figure 4.8: Bi-LSTM network structure for behaviour classification

The Bi-LSTM architecture, illustrated in Fig. 4.8, is employed for behaviour classification. Given an input sequence of state vectors $\mathbf{x}_{t-N:t+N}$, the network processes the trajectory in both temporal directions through a forward LSTM and a backward LSTM. The forward pass captures dependencies from past to present, while the backward pass incorporates contextual information from future observations. Their hidden representations are concatenated and passed through fully connected (or convolutional) layers to produce class logits, which are converted into predictions $\hat{y}_t$. By leveraging bidirectional temporal context, the Bi-LSTM is able to model long-range dependencies and capture subtle behavioural transitions more effectively than a unidirectional LSTM. This makes it particularly suitable for flight behaviour classification, where both preceding and subsequent manoeuvres provide valuable cues for accurate labelling. Furthermore, the sequential nature of these architectures supports near real-time operation, as the model can be deployed with a short sliding window to balance classification latency with predictive accuracy.

4.3 Network Training

In this section, the training procedure of the aforementioned pipeline is described for two main tasks: an unsupervised LSTM-AE for conformance monitoring, and a supervised Bi-LSTM for behaviour classification at each time instant. The training dataset consists exclusively of nominal trajectories that do not include class C_X , with five trajectories exhibiting deviated behaviour excluded from the set. This makes LSTM-AE learn the patterns of the nominal behaviour, so that reconstruction error reflects deviation. The Bi-LSTM is trained on the labelled dataset across all behaviour classes from C_1 to C_5 .

All state variables in $\mathbf{x}_t$ are normalised using z -score statistics computed from the training dataset and consistently applied to the test dataset to prevent data leakage. The dataset is partitioned into 70% training and 30% test sets, with class distributions shown in Fig. 4.9, demonstrating broadly comparable proportions across the split. Ensuring such consistency is crucial, as discrepancies in class distribution between training and test sets would undermine the reliability of the evaluation. For real-time inference, sequences are reformulated into overlapping sliding windows $\mathbf{x}_{t-N:t+N}$ and fed to both models, enabling early anomaly detection while preserving short-term temporal context. Consequently, the 360 trajectories $\{\mathbf{x}_1 : \mathbf{x}_f\}$ are transformed into a three-dimensional tensor of size $(M \times T \times D)$, where $M = 20,519$ denotes the total number of samples, $T = 5$ the sliding-window length $(2N + 1)$, and $D = 7$ the number of state variables.

Tab. 4.2 summarises the hyperparameter configurations employed for training the anomaly detection and behaviour classification networks. Both networks adopt the same input window, feature dimension, hidden layers, optimiser, and training settings for consistency. The LSTM-AE minimises reconstruction error with a latent bottleneck, while the Bi-LSTM predicts one of five classes using cross-entropy loss and softmax outputs.

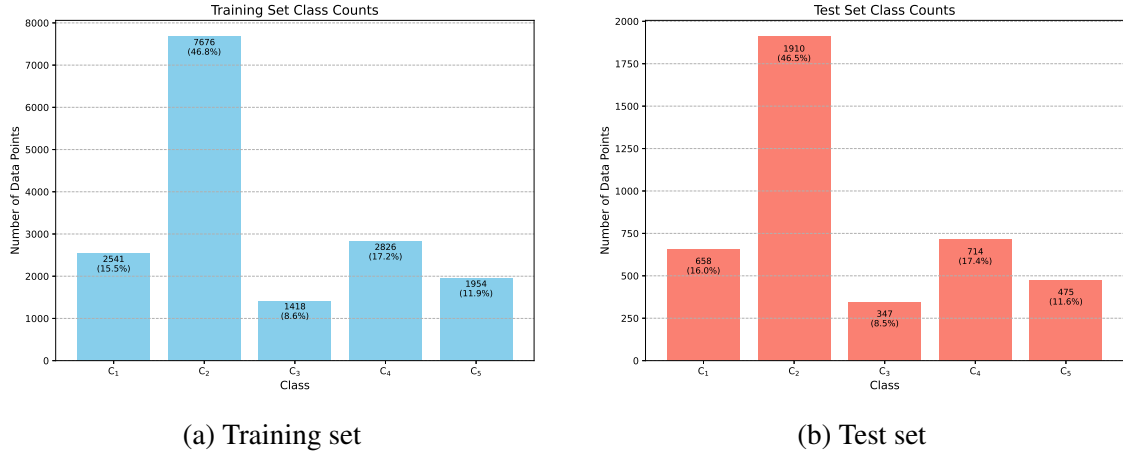


Figure 4.9: Class distribution across training and test datasets

Table 4.2: Network hyperparameters used for anomaly detection (LSTM-AE) and behaviour classification (Bi-LSTM)

Hyperparameter	LSTM-AE	Bi-LSTM
Input window length	$2N+1$ ($N = 2$)	$2N+1$ ($N = 2$)
Feature dimension	$D=7$	$D=7$
Hidden units per layer	16	16
# LSTM layers	2	2
Dropout	0.3	0.3
Bottleneck (latent size)	16	–
Output dimension	D (reconstruction of input)	5 (# classes)
Loss	MSE (reconstruction)	Cross-entropy
Optimiser	Adam	Adam
Learning rate	1×10^{-3}	1×10^{-3}
Batch size	16	16
Epochs	100	100
Normalisation	z -score	z -score
Thresholding / decision	Anomaly score from reconstruction error	Argmax of softmax at time instant t

4.4 Conclusion

In summary, this chapter presented the methodology for developing a real-time framework for conformance monitoring and behaviour classification using ADS-B data. The approach integrates data preprocessing, feature engineering, and behaviour labelling with a two-stage neural network pipeline. The LSTM-AE enables anomaly detection by learning nominal kinematic patterns, while the Bi-LSTM performs sequential classification of flight behaviours by leveraging bidirectional temporal dependencies. Both models are trained with hyperparameters and deployed on sliding windows to support real-time inference. This methodology establishes the foundation for the performance evaluation and experimental results presented in the next chapter.

Chapter 5

Simulation

In this chapter, the proposed frameworks are evaluated through simulations on the ADS-B dataset. First, the conformance monitoring task is addressed using an LSTM-AE-based anomaly detector, with comparisons against a Graph Autoencoder (GAE). The evaluation examines reconstruction error distributions on normal data, anomaly detection performance under deviated behaviours, and the effect of contextual NOTAMs information on anomaly probability. Subsequently, the behaviour classification task is investigated using recurrent and convolutional architectures, and their performance is assessed on the held-out test dataset through confusion matrices, KPIs, Monte-Carlo probability traces, and sliding-window real-time analysis.

5.1 Conformance Monitoring

This section analyses the conformance monitoring performance of the proposed LSTM-AE network, alongside GAE. The LSTM-AE is adopted as the primary model since its recurrent structure is well-suited for sequential reconstruction, allowing it to capture temporal dependencies and deviations in flight trajectories. In contrast, GAE provides a complementary perspective by leveraging graph connectivity and relational features rather than temporal recurrence, enabling anomaly detection through the reconstruction of latent graph structures. Together, these two models highlight the trade-off between time-series based reconstruction and graph-structured representation in detecting anomalous behaviours.

Fig. 5.1 illustrates a conformance monitoring case where an anomalous go-around is detected near KSFO. In the scenario view Fig. 5.1a and Fig. 5.1b, the aircraft initially aligns with runway 28R but aborts its landing due to the closure specified in the NOTAM !SF0 03/112 SF0 RWY 28R CLSD 1803122030–1803122130. The trajectory shows the aircraft entering a hovering manoeuvre before re-approaching and eventually landing on runway 28L. This operational irregularity is reflected in the anomaly scores in Fig. 5.1c and Fig. 5.1d. The anomalous behaviour initiates at time instant 21:18, marking the onset of the go-around manoeuvre. At 21:20, although the aircraft returns to a nominal flight path, the model still interprets this transition as anomalous due to the sudden phase change. The LSTM-AE captures this with an anomaly score peaking around 0.6, whereas the GAE does not register it as anomalous at that point. Instead, the GAE highlights the drastic turn occurring at 21:28 as an anomaly, reflecting its sensitivity to

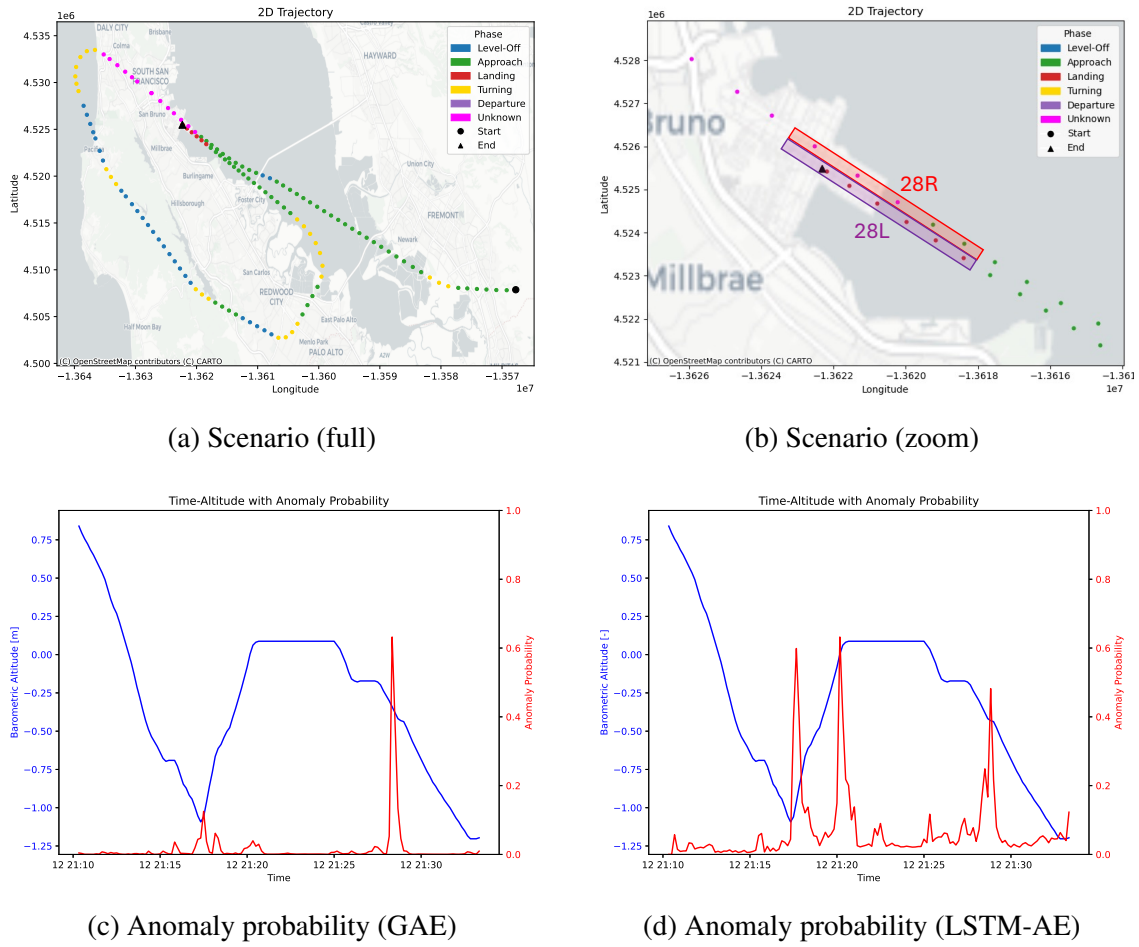


Figure 5.1: Conformance monitoring results: scenario views (top) and anomaly probabilities from GAE and LSTM-AE (bottom)

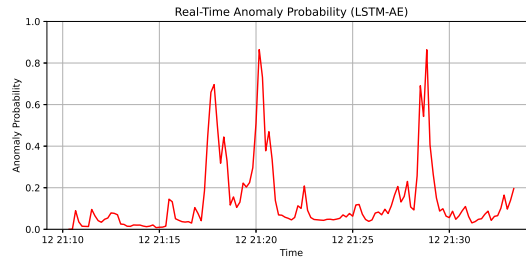


Figure 5.2: NOTAMs induced anomaly probability (LSTM-AE)

relational dynamics rather than short-term reconstruction deviations.

Incorporating NOTAM context, the design parameter α in (3.11) serves as a calibration weight that maps the normalised reconstruction error to an anomaly probability. During NOTAM-active intervals that prohibit specific behaviours (e.g., the 28R closure), α can be increased (e.g., from 1 to 2) to up-weight such violations, resulting in higher anomaly scores. As shown in Fig. 5.2, this adjustment makes the go-around behaviour more clearly exceed the threshold and be flagged as anomalous. Importantly, this modification influences only the probability mapping, while LSTM-AE model unchanged.

5.2 Behaviour Classification

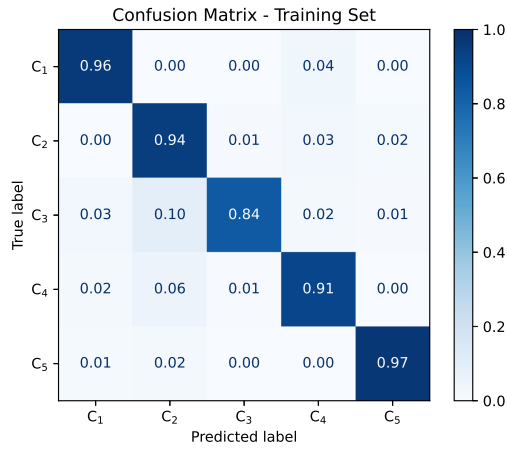
This section analyses the behaviour classification performance of the proposed Bi-LSTM network, alongside two baseline models, LSTM and 1D-CNN. The LSTM is selected as a benchmark because it represents the most widely adopted recurrent neural network architecture for sequential modelling, capable of capturing temporal dependencies in flight trajectories. The 1D-CNN is included as a contrasting architecture that processes sequences using convolutional filters rather than recurrent connections, providing strong local feature extraction while being computationally efficient.

All three networks are trained under identical hyperparameter settings, including normalisation scheme, Adam optimiser, learning rate, batch size, dropout, and epochs. While the recurrent models are parameterised by the number of hidden units and layers, the 1D-CNN is instead configured with 32 convolutional filters and a kernel size of 3, which serve as the equivalent architectural parameters. The classification results on the training and test datasets are illustrated in Fig. 5.3 and Fig. 5.4, which present the confusion matrices alongside the KPIs for each class. It should be noted that accuracy is not class-specific, hence the reported values are identical across all classes.

The training results highlight clear differences across the three network architectures. From the confusion matrices from Fig. 5.3a to Fig. 5.3c, most of the true positive rates TP_C exceed 0.80, indicating that the networks generally capture class-specific patterns well. A recurring misclassification trend is the false alarm of other behaviours to *Approach* (C_2). For instance, the LSTM incorrectly predicts *Landing*, *Level-Off*, and *Turning* as *Approach* with probabilities of 0.10, 0.06, and 0.02, respectively, while the 1D-CNN exhibits similar confusions of 0.09, 0.10, and 0.05. In contrast, the Bi-LSTM substantially reduces these errors, with only a 0.05 false prediction of *Landing* as *Approach*, demonstrating its superior ability to disambiguate closely related manoeuvres.

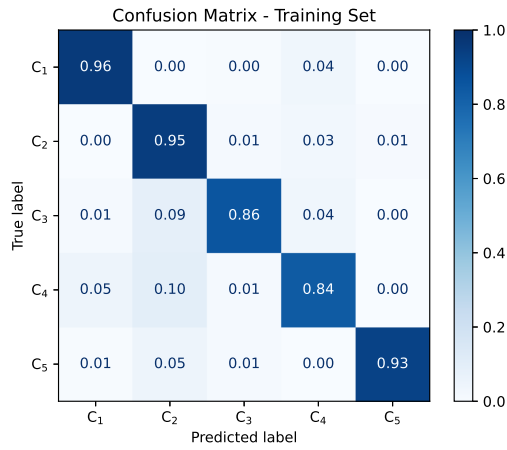
The per-class KPIs further reinforce these findings. While all models achieve high performance, the LSTM and 1D-CNN show relatively lower precision, recall, and F1-scores of around 0.90 for C_3 – C_5 (*Level-Off*, *Turning*, *Landing*), reflecting the challenge of distinguishing these short or transitional phases. The Bi-LSTM consistently outperforms the other models across all metrics, highlighting its strength in recognising temporal dependencies. The only exception is a marginally lower recall for C_5 (*Landing*), suggesting that while Bi-LSTM is robust in avoiding false alarms, a small fraction of actual landings remain under-detected. Overall, the results indicate that the Bi-LSTM provides the most reliable behaviour classification during training, with reduced confusion in transitional phases compared to the LSTM and 1D-CNN.

The test dataset performance, illustrated in Fig. 5.4, remains broadly consistent with the training results. The overall accuracy for all three models lies in the range of 92–98%, confirming that the networks generalise well without evident overfitting. Similar to the training dataset, a notable misclassification pattern is observed as the false alarm of C_2 (*Approach*), where *Landing*, *Level-Off*, and *Turning* are occasionally predicted as *Approach*. For instance, LSTM and 1D-CNN show false alarm rates of up to 0.11 in these classes, while the Bi-LSTM substantially suppresses such errors, reducing them to less than 0.04. In terms of KPIs, C_3 – C_5 (*Level-Off*, *Turning*, and *Landing*) again exhibit relatively lower precision, recall, and F1-scores in the range of 0.83–0.97 for LSTM and 1D-CNN, reflecting the intrinsic difficulty of distinguishing between manoeuvring be-



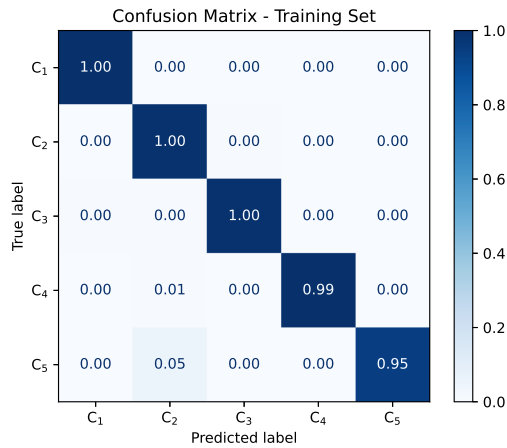
Class	Acc	Prec	Rec	F1
C_1	0.94	0.95	0.96	0.96
C_2		0.96	0.95	0.95
C_3		0.92	0.84	0.88
C_4		0.89	0.92	0.90
C_5		0.93	0.97	0.95

(a) LSTM



Class	Acc	Prec	Rec	F1
C_1	0.92	0.93	0.96	0.95
C_2		0.94	0.95	0.94
C_3		0.87	0.86	0.86
C_4		0.86	0.83	0.85
C_5		0.96	0.93	0.94

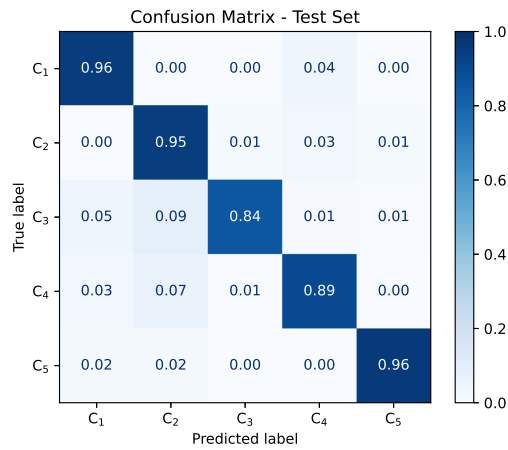
(b) 1D-CNN



Class	Acc	Prec	Rec	F1
C_1	0.99	1.00	0.99	0.99
C_2		0.98	0.99	0.99
C_3		0.98	0.99	0.98
C_4		0.99	0.99	0.99
C_5		0.98	0.94	0.96

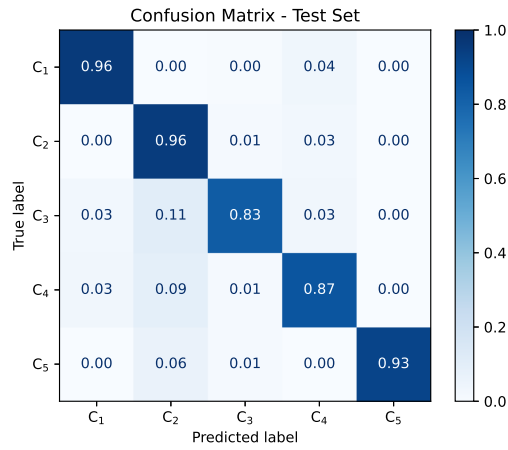
(c) Bi-LSTM

Figure 5.3: Behaviour classification performance on the training dataset; confusion matrix (left) and per-class precision, recall, F1-score, and accuracy (right)



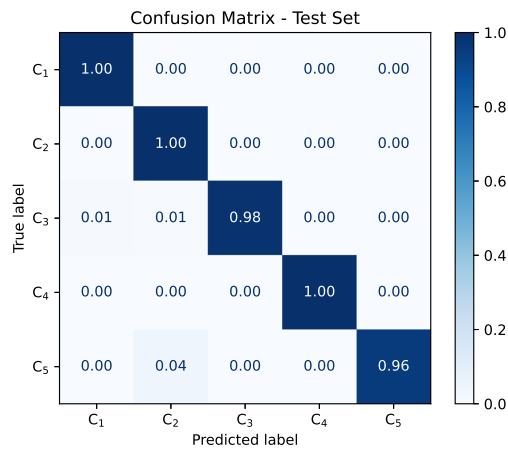
Class	Acc	Prec	Rec	F1
C_1	0.92	0.93	0.94	0.94
C_2		0.95	0.94	0.95
C_3		0.91	0.83	0.87
C_4		0.85	0.89	0.87
C_5		0.93	0.95	0.94

(a) LSTM



Class	Acc	Prec	Rec	F1
C_1	0.92	0.94	0.96	0.95
C_2		0.94	0.95	0.94
C_3		0.89	0.87	0.88
C_4		0.86	0.84	0.85
C_5		0.97	0.93	0.95

(b) 1D-CNN



Class	Acc	Prec	Rec	F1
C_1	0.98	0.99	0.99	0.99
C_2		0.98	0.98	0.98
C_3		0.96	0.96	0.96
C_4		0.97	0.98	0.97
C_5		0.97	0.94	0.96

(c) Bi-LSTM

Figure 5.4: Behaviour classification performance on the test dataset; confusion matrix (left) and per-class precision, recall, F1-score, and accuracy (right)

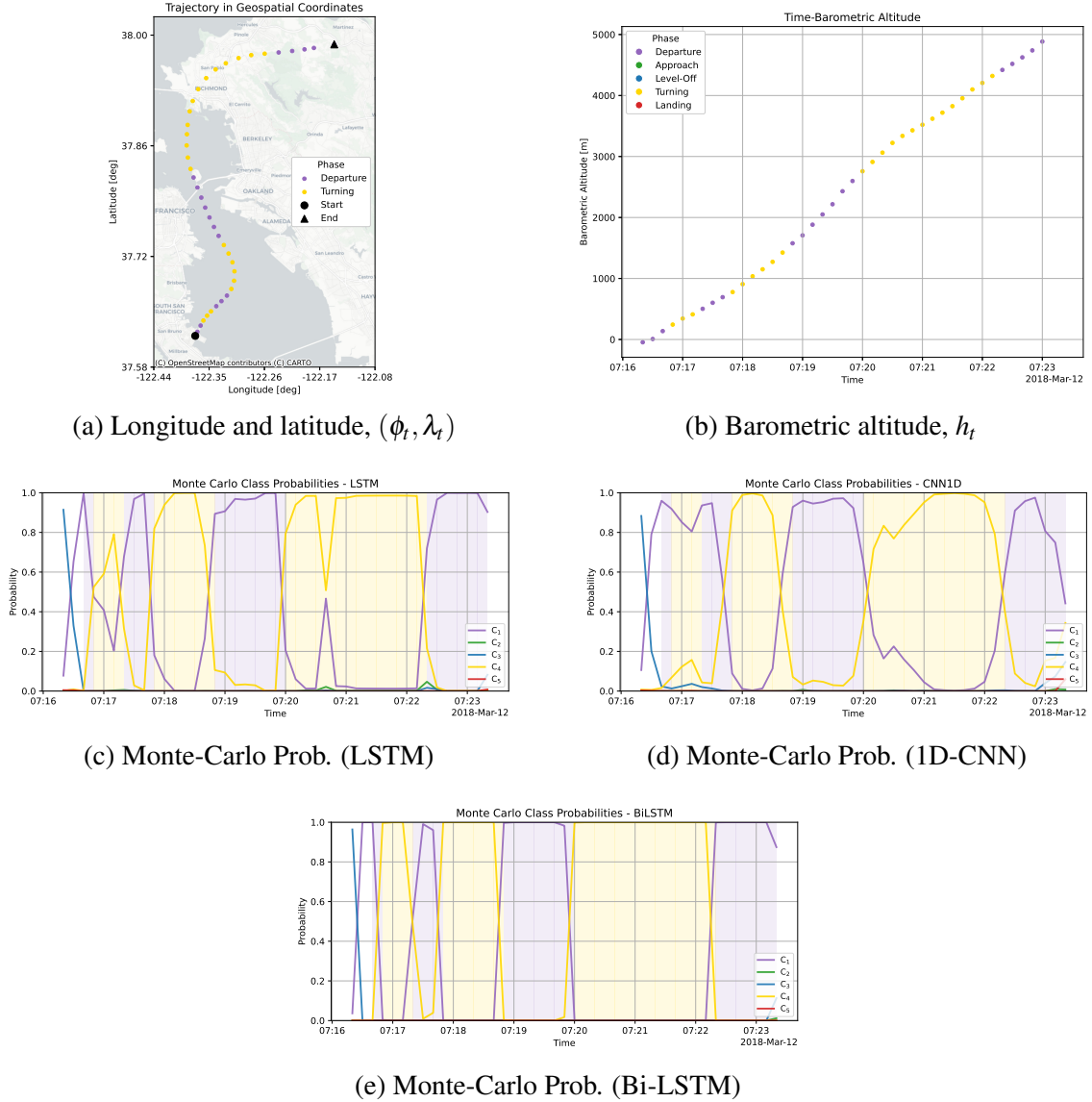


Figure 5.5: Behaviour classification results for the departure scenario

haviours. The Bi-LSTM consistently outperforms both baselines, exceeding 0.96 in most metrics and only slightly underperforming in $Rec(C_5)$ (*Landing*). This consistency across training and test datasets demonstrates the robustness and generalisation capability of the Bi-LSTM network.

The real-time performance is further evaluated through Monte-Carlo (MC) simulations, where classification is assessed not only in terms of hard decisions using the MAP (3.12), but also by analysing the softmax posterior probabilities (3.14). This probabilistic perspective is crucial since it reveals model confidence and temporal consistency, which are not captured by hard decisions. To account for randomness, dropout masks are resampled at each iteration, and the MC result is obtained by averaging over 100 iterations. In the probability plots in Fig. 5.5 and Fig. 5.6, the semi-transparent coloured bands indicate the ground-truth phase at each time step, while the coloured curves ($C_1 - C_5$) show the class posterior probabilities averaged across the MC iterations.

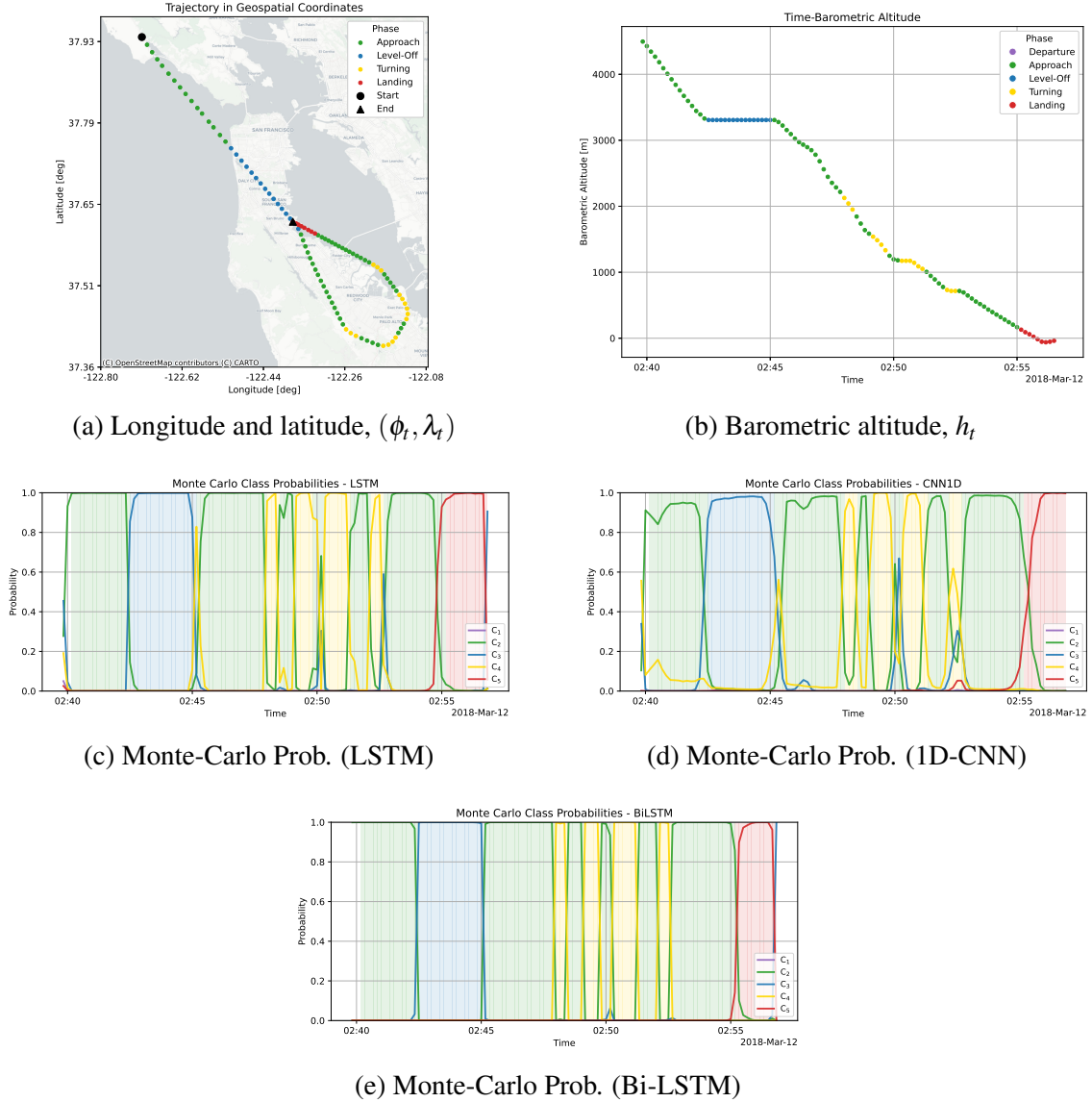


Figure 5.6: Behaviour classification results for the arrival scenario

With a centred window of $N = 2$ and a sampling period $T_s = 10$ s, the classification result exhibits a latency of 20 seconds. In the departure case, the LSTM and 1D-CNN approach exhibit reduced confidence and occasional ambiguity between *Departure* and *Turning*, because turns during climb couple a positive vertical rate with a non-zero heading rate, which the features shared by both classes. By contrast, the Bi-LSTM maintains higher and more stable class probabilities, reflecting its stronger use of temporal context on both sides of the decision point.

For the arrival scenario Fig. 5.6, the same interaction between vertical rate and heading rate changes leads to confusion between *Approach* and *Turning*. Around time instant 02:50, the LSTM and 1D-CNN show oscillatory, low-consistency probabilities, whereas the Bi-LSTM remains confident and produces sharper, better-aligned transitions. These dropout-based Monte-Carlo probability traces complement hard MAP decisions by exposing confidence, transition sharpness, and robustness to stochastic perturbations.

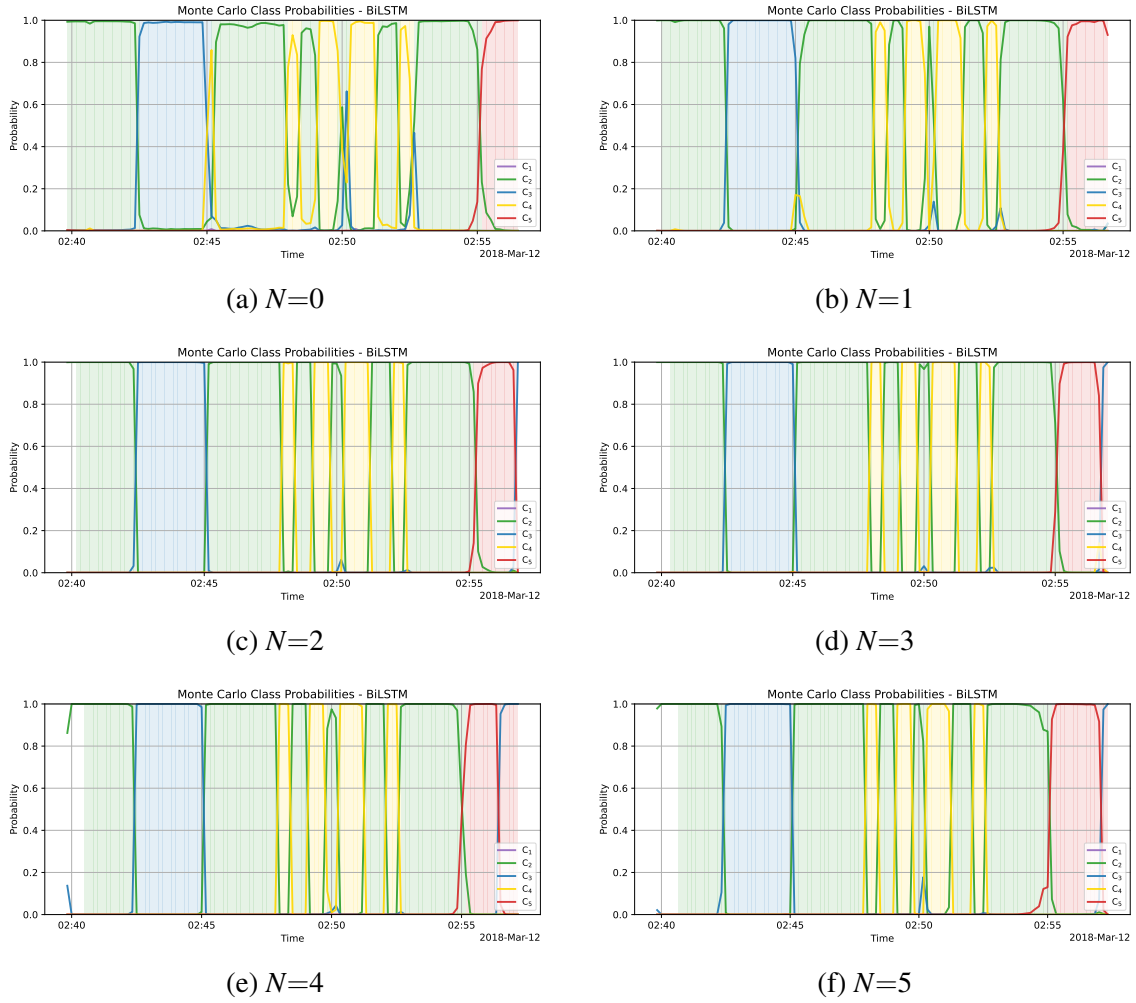


Figure 5.7: Monte-Carlo class probabilities for Bi-LSTM across window size

Fig. 5.7 demonstrates the influence of the sliding-window length N on the real-time behaviour classification performance on the same scenario of Fig. 5.6. When $N = 0$, temporal correlations are not incorporated, and the Bi-LSTM fails to reliably distinguish phase transitions. For example, illustrated in Fig. 5.7a, at time instant 02:50 the model cannot resolve the correct behaviour, as the instantaneous features alone are insufficient. Increasing to $N = 1$ introduces a minimal temporal context, leading to marginal improvements, yet the predictions remain unstable and sensitive to local fluctuations. From $N = 2$, adopted as the proposed configuration in this study, the classification becomes more stable and consistent, with transitions captured accurately and without undue delay. Beyond this point, however, a clear trade-off emerges that while larger N values strengthen temporal smoothing, they inevitably increase latency. At $N = 5$, illustrated in Fig. 5.7f, this latency manifests as delayed phase transitions, such as the shift from *Approach* to *Landing* at 02:55. Overall, $N = 2$ represents an effective compromise, providing sufficient temporal awareness for robust classification while maintaining real-time responsiveness.

5.3 Conclusion

This chapter demonstrated the real-time feasibility of the proposed dual-pipeline framework for conformance monitoring and behaviour classification using ADS-B data. For conformance monitoring, the LSTM-AE effectively captured temporal deviations in flight trajectories, successfully detecting a go-around anomaly induced by a runway closure NOTAM, while the GAE provided complementary sensitivity to relational dynamics. For behaviour classification, the Bi-LSTM consistently outperformed baseline LSTM and 1D-CNN models, particularly in reducing false alarms to the *Approach* class and improving recognition of transitional phases such as Level-Off, Turning, and Landing.

Monte-Carlo simulations confirmed the robustness of the Bi-LSTM in real-time operation, maintaining stable confidence across phase transitions with a sliding window length of $N = 2$, which balanced classification accuracy and latency. Overall, the LSTM-AE proved effective for anomaly detection, while the Bi-LSTM delivered superior classification performance, highlighting the strength of the proposed dual-pipeline approach for supporting resilient, real-time situational awareness in air traffic monitoring.

Chapter 6

Conclusion

6.1 Summary of Contributions

This thesis makes the following primary contributions:

- **Dual-Pipeline Framework:** Proposed a real-time architecture that combines an unsupervised LSTM-AE for conformance monitoring with a supervised Bi-LSTM for behaviour classification. This unified approach enables simultaneous anomaly detection and recognition of dynamic flight phases.
- **Sliding-Window Mechanism for Real-Time Inference:** Designed and evaluated a centred sliding-window strategy to balance accuracy and latency. The chosen configuration ($N = 2$) achieved stable classification and reliable phase transitions with only a 20-second delay, demonstrating real-time feasibility.
- **Context-Aware Extension:** Demonstrated how external operational information, such as NOTAMs, can be incorporated as calibration factors in anomaly scoring, thereby enhancing the interpretability and operational relevance of detection results.

The emphasis on the dual-pipeline design and sliding-window mechanism establishes a foundation for robust, real-time monitoring and classification of aircraft behaviours using ADS-B data.

6.2 Key Findings and Limitations

This research has produced several notable findings. First, the proposed framework demonstrated real-time feasibility, achieving low-latency inference suitable for operational conformance monitoring. Second, the anomaly detection methods proved sensitive to abnormal deviations, even under subtle variations in trajectory dynamics. Third, the integration of ATC-related information, particularly NOTAMs, highlighted the potential of context-aware anomaly detection. Finally, the behaviour classification framework effectively handled class distribution, maintaining robust performance across major operational phases.

Despite these strengths, certain limitations should be acknowledged. The dataset suffered from data and label scarcity, limiting generalisability and the ability to capture rare

manoeuvres. The behaviour taxonomy was restricted to a small set of operational phases, excluding more complex behaviours encountered in practice. The framework was also primarily evaluated in a civil aviation context, with limited attention to multi-domain or military operations. Moreover, while NOTAMs were considered as ATC context, other sources of operational information (e.g., flight plans, weather reports, or sector capacity constraints) were not yet integrated, but could enhance performance in future work.

6.3 Recommendations and Future Work

This research has demonstrated the feasibility of real-time conformance monitoring and behaviour classification using ADS-B data. Nevertheless, several directions remain open for further investigation.

- The current study focused on a limited set of operational phases. Future work may extend the taxonomy to include more complex behaviours such as zig-zag manoeuvres, box patterns, patrols, holds, and vectoring. Achieving this requires access to sufficient non-commercial and military flight datasets, along with unsupervised or semi-supervised clustering methods to automatically derive reliable labels at each time instant.
- For scenarios involving multiple cooperative or conflicting agents, transformer-based architectures could be explored. Such models are well-suited to capture spatio-temporal dependencies across multiple trajectories, potentially integrating anomaly detection and behaviour classification into a unified framework with high-dimensional ADS-B inputs.
- Incorporating contextual information such as weather data, airspace restrictions, and NOTAMs can enhance both anomaly detection and behaviour classification by embedding environmental constraints into the decision process.
- To bridge the gap between simulation and deployment, future studies should validate the proposed framework against real operational data streams (e.g., live ADS-B feeds) and evaluate performance under latency, communication loss, and adversarial conditions.
- Beyond reconstruction error, hybrid anomaly detection strategies combining statistical tests, probabilistic graphical models, and graph neural networks could be explored to better capture rare but safety-critical events.
- Further optimisation of model architectures and inference pipelines is necessary to ensure scalability to continental or global-scale traffic monitoring without sacrificing latency requirements.

References

1. Training UA, Command D. The US Army in Multi-Domain Operations 2028 (TRADOC Pamphlet 525-3-1). United States Army. 2018.
2. Habler E, Shabtai A. Using LSTM encoder-decoder algorithm for detecting anomalous ADS-B messages. *Computers & Security*. 2018;78:155-73.
3. Memarzadeh M, Matthews B, Avrekh I. Unsupervised anomaly detection in flight data using convolutional variational auto-encoder. *Aerospace*. 2020;7(8):115.
4. Zhang X, Li J. SR-GNN: Spectral Residuals with Graph Neural Networks for Anomaly Detection in ADS-B Data. *International Conference on Frontiers in Cyber Security*. Springer; 2023. p. 503-16.
5. Ester M, Kriegel HP, Sander J, Xu X, et al. A density-based algorithm for discovering clusters in large spatial databases with noise. *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining (KDD)*. AAAI Press; 1996. p. 226-31.
6. Campello RJ, Moulavi D, Sander J. Density-based clustering based on hierarchical density estimates. *Pacific-Asia conference on knowledge discovery and data mining*. Springer; 2013. p. 160-72.
7. Kim H, Lee BS, Shin WY, Lim S. Graph anomaly detection with graph neural networks: Current status and challenges. *IEEE Access*. 2022;10:111820-9.
8. Reynolds TG, Hansman RJ. Investigating conformance monitoring issues in air traffic control using fault detection techniques. *Journal of aircraft*. 2005;42(5):1307-17.
9. Bishop CM, Nasrabadi NM. *Pattern recognition and machine learning*. Springer; 2006.
10. Blázquez-García A, Conde A, Mori U, Lozano JA. A review on outlier/anomaly detection in time series data. *ACM computing surveys (CSUR)*. 2021;54(3):1-33.
11. Wang J, Zou Y, Ding J. ADS-B spoofing attack detection method based on LSTM. *EURASIP Journal on Wireless Communications and Networking*. 2020;2020(1):160.
12. Chevrot A, Vernotte A, Legeard B. DAE: Discriminatory Auto-Encoder for Multivariate Time-Series Anomaly Detection in Air Transportation. *arXiv preprint arXiv:210904247*. 2021.

13. Brooks TN. Using autoencoders to learn interesting features for detecting surveillance aircraft. arXiv preprint arXiv:180910333. 2018.
14. Campbell N, Grauer JA, Gregory IM. Use of design of experiments in determining neural network architectures for loss of control detection. IEEE Aerospace Conference; 2021. .
15. Luo P, Wang B, Li T, Tian J. ADS-B anomaly data detection model based on VAE-SVDD. Computers & Security. 2021;104:102213.
16. Habler E, Shabtai A. Analyzing sequences of airspace states to detect anomalous traffic conditions. IEEE Transactions on Aerospace and Electronic Systems. 2021;58(3):1843-57.
17. Yang Z, Kang X, Gong Y, Wang J. Aircraft trajectory prediction and aviation safety in ADS-B failure conditions based on neural network. Scientific reports. 2023;13(1):19677.
18. Jin M, Koh HY, Wen Q, Zambon D, Alippi C, Webb GI, et al. A survey on graph neural networks for time series: Forecasting, classification, imputation, and anomaly detection. IEEE Transactions on Pattern Analysis and Machine Intelligence. 2024.
19. Wang X, Ma Y, Wang Y, Jin W, Wang X, Tang J, et al. Traffic flow prediction via spatial temporal graph neural network. Proceedings of the web conference 2020; 2020. p. 1082-92.
20. Chen W, Tian L, Chen B, Dai L, Duan Z, Zhou M. Deep variational graph convolutional recurrent network for multivariate time series anomaly detection. International conference on machine learning. PMLR; 2022. p. 3621-33.
21. Han S, Woo SS. Learning sparse latent graph representations for anomaly detection in multivariate time series. Proceedings of the 28th ACM SIGKDD Conference on knowledge discovery and data mining; 2022. p. 2977-86.
22. Deng A, Hooi B. Graph neural network-based anomaly detection in multivariate time series. Proceedings of the AAAI conference on artificial intelligence. vol. 35; 2021. p. 4027-35.
23. Chen Z, Chen D, Zhang X, Yuan Z, Cheng X. Learning graph structures with transformer for multivariate time-series anomaly detection in IoT. IEEE Internet of Things Journal. 2021;9(12):9179-89.
24. Zheng Y, Koh HY, Jin M, Chi L, Phan KT, Pan S, et al. Correlation-aware spatial-temporal graph learning for multivariate time-series anomaly detection. IEEE transactions on neural networks and learning systems. 2023;35(9):11802-16.
25. Zheng Y, Koh HY, Jin M, Chi L, Wang H, Phan KT, et al. Graph spatiotemporal process for multivariate time series anomaly detection with missing values. Information Fusion. 2024;106:102255.

26. Zhao H, Wang Y, Duan J, Huang C, Cao D, Tong Y, et al. Multivariate time-series anomaly detection via graph attention network. 2020 IEEE international conference on data mining (ICDM). IEEE; 2020. p. 841-50.
27. Chen K, Feng M, Wirjanto TS. Multivariate time series anomaly detection via dynamic graph forecasting. arXiv preprint arXiv:230202051. 2023.
28. Kingma DP, Rezende DJ, Mohamed S, Welling M. Semi-supervised learning with deep generative models. Advances in neural information processing systems. 2014;27.
29. Memarzadeh M, Matthews B, Templin T. Multi-Class anomaly detection in flight data using semi-supervised explainable deep learning model. Journal of Aerospace Information Systems. 2022;19(2):83-97.
30. Karam R, Salomon M, Couturier R. Supervised ADS-B anomaly detection using a false data generator. International Conference on Computer, Control and Robotics; 2022. .
31. Bosson CS, Nikoleris T. Supervised learning applied to air traffic trajectory classification. 2018 AIAA Information Systems-AIAA Infotech@ Aerospace; 2018. p. 1637.
32. Lee L, Fontes J, Weinert A, Schomacker L, Stabile D, Hou J. TAACKIT: Track Annotation and Analytics with Continuous Knowledge Integration Tool. 2024 International Conference on AI x Data and Knowledge Engineering (AIXDKE). IEEE; 2024. p. 112-6.
33. Minev MA. Motion Dynamics Based Aircraft Trajectory Classification Using Neural Networks. Master's thesis. Prague, Czech Republic: Czech Technical University in Prague, Faculty of Transport Sciences; 2022. Department of Air Transport.
34. Zhang S, Zhang Y, Tay T, Shankar J. Learning-based aircraft trajectory analysis tool for holding and vectoring identification with ADS-B data. 2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC). IEEE; 2022. p. 1100-5.
35. Kim K, Hwang I, Yang BJ. Classification of conflict resolution methods using data-mining techniques. 16th AIAA Aviation Technology, Integration, and Operations Conference; 2016. p. 4075.
36. Sun J, Ellerbroek J, Hoekstra J. Large-scale flight phase identification from ads-b data using machine learning methods. 7th International Conference on Research in Air Transportation; 2016. p. 1-8.
37. Arts E. A Novel Approach to Flight Phase Identification using Machine Learning. Universität Hamburg; 2021.
38. Arts E, Kamtsiuris A, Meyer H, Raddatz F, Peters A, Wermter S. Trajectory Based Flight Phase Identification with Machine Learning for Digital Twins. Deutsche Gesellschaft für Luft- und Raumfahrt - Lilienthal-Oberth eV. 2022.

39. Deng C, Choi HC, Park H, Hwang I. Trajectory pattern identification and classification for real-time air traffic applications in Area Navigation terminal airspace. *Transportation Research Part C: Emerging Technologies*. 2022;142:103765.
40. Zhang M, Zhang L, Liu T. Aircraft Behaviour Recognition on Trajectory Data with a Multimodal Approach. *Electronics*. 2024;13(2):367.
41. Blythe W, Anderson H, King N, et al. Ads-b implementation and operations guidance document. International Civil Aviation Organization Asia and Pacific Ocean. 2011.
42. Federal Aviation Administration. Ins and Outs: Capabilities of the ADS-B System; 2023. https://www.faa.gov/air_traffic/technology/equipadsb/capabilities/ins_outs.
43. Federal Aviation Administration. Automatic Dependent Surveillance–Broadcast (ADS–B) Out, In, and Ground Stations: System Overview. Federal Aviation Administration. 2023.
44. EUROCONTROL. ATM Surveillance Tracker and Server (ARTAS); 2025. <https://www.eurocontrol.int/product/artas>. Processing ADS-B, radar, Mode-S and multilateration data; deployed in over 31 ATC centres in Europe.
45. Aireon LLC. Space-Based ADS-B: Global Real-Time Aircraft Tracking; 2024. <https://aireon.com/ads-b-for-air-traffic-surveillance/>. Satellite-based ADS-B providing global real-time flight tracking, including remote and oceanic regions.
46. Federal Aviation Administration. Capstone Program: ADS-B Demonstration in Rural Alaska; 2006. https://www.faa.gov/air_traffic/technology/adsb/archival. Joint FAA–industry safety demonstration deploying ADS-B avionics and ground infrastructure in rural Alaska.
47. International Civil Aviation Organization. Procedures for Air Navigation Services — Air Traffic Management (Doc 4444), 16th ed.; 2016. ICAO Doc 4444. Available at: <https://www.icao.int>. Effective 10 November 2016.
48. International Civil Aviation Organization. Global Operational Data Link Document (GOLD); 2018. ICAO Doc 10037.
49. Federal Aviation Administration. Traffic Flow Management in the National Airspace System. Federal Aviation Administration, Air Traffic Organization; 2009. Available at: https://www.fly.faa.gov/Products/Training/Traffic_Management_for_Pilots/TFM_in_the_NAS_Booklet_ca10.pdf. A presentation explaining terms, techniques, and programs associated with traffic flow management in the National Airspace System.
50. EUROCONTROL. Network Manager Business-to-Business (B2B) Concept of Operations; 2024. Concept of Operations and Reference Manual. Available at: <https://www.eurocontrol.int/publication/network-manager-b2b-concept-operations>.

51. International Civil Aviation Organization. Annex 15 — Aeronautical Information Services, 16th ed.; 2018. Chicago Convention Annex 15. Available at: <https://store.icao.int/en/annexes/annex-15>.
52. International Civil Aviation Organization. Doc 10066 — Procedures for Air Navigation Services: Aeronautical Information Management (PANS–AIM); 2018. ICAO Doc 10066. Available at: <https://store.icao.int/en/aeronautical-information-management-doc-10066>.
53. AIXM Community. AIXM 5.1.1 Temporality Concept. EUROCONTROL/FAA AIXM; 2019. Available at: https://aixm.aero/sites/default/files/imce/AIXM511/aixm_temporality_1.1.pdf.
54. Federal Aviation Administration. System Wide Information Management (SWIM) Program Overview; 2025. Online. Available at: https://www.faa.gov/air_traffic/technology/swim/overview.
55. Olive X. Traffic, a toolbox for processing and analysing air traffic data. *Journal of Open Source Software*. 2019;4:1518.
56. Naganawa J, Miyazaki H. A method for accurate ADS-B signal strength measurement under co-channel interference. 2018 Asia-Pacific Microwave Conference (APMC). IEEE; 2018. p. 354-6.
57. Purton L, Abbass H, Alam S. Identification of ADS-B system vulnerabilities and threats. Australian Transport Research Forum, Canberra. sn; 2010. p. 1-16.
58. International Civil Aviation Organization (ICAO). Annex 10 — Aeronautical Telecommunications, Volume II: Communication Procedures including those with PANS status. Montréal, Canada: International Civil Aviation Organization; 2016. Chicago Convention Annex 10, Volume II.
59. Maaten Lvd, Hinton G. Visualizing data using t-SNE. *Journal of machine learning research*. 2008;9(Nov):2579-605.