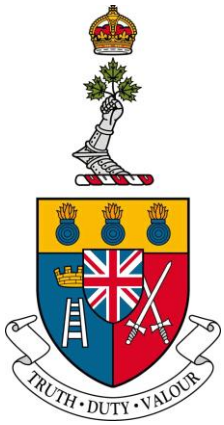
COMBINING MACHINE LEARNING AND GEOSPATIAL ANALYSIS FOR A BETTER UNDERSTANDING OF COMPLEX AIRSPACE ENVIRONMENTS

FUSION DE L'INTELLIGENCE ARTIFICIELLE ET DE L'ANALYSE GÉOSPATIALE POUR UNE MEILLEURE COMPRÉHENSION DES ESPACES AÉRIENS COMPLEXES



A Thesis Submitted to the Division of Graduate Studies
of the Royal Military College of Canada
by

Nicolas Vincent-Boulay

In Partial Fulfillment of the Requirements for the Degree of
Doctor of Philosophy

October, 2026

© This thesis may be used within the Department of National Defence, but copyright for open publication remains the property of the author

Acknowledgments

Looking back on the years that have led to the completion of this thesis, I realize that pursuing a PhD has been one of the most challenging, rewarding, and intellectually enriching experiences of my life. Beyond the research itself, this journey has provided an invaluable opportunity for personal and professional growth, shaped in no small part by the remarkable people I have been fortunate enough to meet along the way.

First and foremost, I would like to express my deepest gratitude to my supervisor, Professor Catharine Marsden, for her invaluable guidance, patience, and encouragement throughout this journey. Her thoughtful advice, dedication, and commitment to scientific rigour have helped shape both this research and my development as a researcher and critical thinker. Of the many lessons I will take away from this experience, perhaps the most important is the value of perseverance: that even the most difficult challenges can be overcome through patience, curiosity, and the determination not to give up. I consider myself extremely fortunate to have benefited from her mentorship, and I sincerely hope that other students will have the opportunity to learn from someone equally dedicated and supportive.

I am also deeply grateful to my family for their unconditional love, encouragement, and understanding throughout these years. Their unwavering support has been a constant source of motivation, particularly during the more demanding periods of this journey. Although this thesis represents years of individual work, its completion would not have been possible without the people who have stood beside me throughout it.

To my fellow graduate students and friends, thank you for the countless conversations, encouragement, and memorable moments along the way. The opportunity to exchange ideas, learn from one another, and share both the successes and frustrations of research has made this experience considerably more enjoyable and meaningful. I will remember not only the work we accomplished but also the friendships formed throughout these years.

Finally, I would like to sincerely thank Marinvent Corporation and Mitacs for their financial support of this research. I am particularly grateful to Marinvent for its continued collaboration and for sharing valuable subject-matter expertise and practical aerospace engineering perspectives at various stages of the project. Their support and guidance contributed meaningfully to the successful completion of this work. I would also like to acknowledge the OpenSky Network community and the National Oceanic and Atmospheric Administration (NOAA), whose commitment to making scientific data openly available was essential to this research.

I will always look back on this journey with gratitude, not only for the knowledge and experience I have gained, but especially for the people who made it such a meaningful chapter of my life.

Abstract

The growing availability of large historical aerospace datasets, together with rising air traffic volumes and the emergence of new air vehicle types and operations, creates new opportunities for data-driven analysis of increasingly complex airspace environments. However, aircraft surveillance, weather, and other aerospace datasets differ substantially in spatial structure, temporal resolution, physical meaning, and data volume, complicating their integration and reuse across analytical applications. This thesis develops and evaluates a modular, reusable, data-driven four-dimensional airspace-modelling methodology that leverages large historical aerospace datasets while managing the associated computational requirements.

The methodology was implemented using historical Automatic Dependent Surveillance–Broadcast (ADS-B) aircraft trajectories from the OpenSky Network and NOAA NEXRAD weather radar data over the New York metropolitan airspace. A Discrete Global Grid System (DGGS) was used to associate heterogeneous observations with common three-dimensional locations and time references. Three experiments evaluated the methodology. Experiment #1 developed representative DGGS-based airspace models and evaluated feature selection, dataset partitioning, and model generation and reuse, while verifying that the model information required by downstream analyses was preserved. Experiment #2 evaluated analytical reuse of the model through a location-based risk case study in which weather risk and midair-collision (MAC) risk indicators were independently derived and subsequently combined. Experiment #3 evaluated data-driven identification and interpretation of recurring operational patterns from trajectory-level airspace-model outputs using a selected KMeans implementation, category-granularity analysis, and comparison with independent aircraft-reference information.

The results show that heterogeneous aircraft and weather observations can be integrated into reusable DGGS-indexed 4D model outputs while preserving the information required for subsequent analyses. Feature selection reduced model-generation time and saved output size, and dataset partitioning substantially reduced memory demand at the cost of additional processing time. Reusing previously generated model outputs enabled substantially faster repeated analyses. The risk experiment demonstrated that independently derived aircraft- and weather-related indicators can be co-indexed, interpreted, and combined within the same spatial and temporal reference. The clustering experiment identified recurring terminal-area and higher-altitude trajectory patterns without predefined behavioural labels. Post-clustering spatial and kinematic evidence supported their interpretation as operationally meaningful behaviour categories. Comparison of the $K = 5$ and $K = 9$ solutions demonstrated adjustable category granularity, and independent aircraft-reference information showed that operationally distinct behaviour categories can share the same dominant conventional aircraft classification.

Together, the experiments demonstrate a reusable, data-driven airspace-modelling framework that separates the persistent representation of heterogeneous historical airspace data from the application-specific analyses subsequently applied to it. The methodology supports computationally practical model generation and reuse, the derivation and combination of multiple location-based analytical products, data-driven identification and post-clustering interpretation of recurring operational patterns, and visualization-supported analysis of complex 4D airspace environments.

Résumé

La disponibilité croissante de vastes ensembles de données aérospatiales historiques, combinée à l'augmentation du volume du trafic aérien et à l'émergence de nouveaux types d'aéronefs et d'opérations, crée de nouvelles possibilités d'analyse fondée sur les données dans des environnements d'espace aérien de plus en plus complexes. Toutefois, les données de surveillance des aéronefs, les données météorologiques et les autres données aérospatiales diffèrent considérablement quant à leur structure spatiale, leur résolution temporelle, leur signification physique et leur volume, ce qui complique leur intégration et leur réutilisation dans différentes applications analytiques. Cette thèse développe et évalue une méthodologie modulaire, réutilisable et fondée sur les données pour la modélisation quadridimensionnelle de l'espace aérien à partir de vastes ensembles de données aérospatiales historiques, tout en gérant les exigences de calcul qui leur sont associées.

La méthodologie a été mise en œuvre à partir de trajectoires historiques d'aéronefs provenant de la surveillance dépendante automatique en mode diffusion (ADS-B) du réseau OpenSky et de données de radar météorologique NEXRAD de la NOAA couvrant l'espace aérien de la région métropolitaine de New York. Un système de grille globale discrète (DGGS) a été utilisé afin d'associer des observations hétérogènes à des positions tridimensionnelles et à des références temporelles communes. Trois expériences ont permis d'évaluer la méthodologie. La première expérience a permis de développer des modèles représentatifs de l'espace aérien fondés sur un DGGS et d'évaluer la sélection de variables, le partitionnement des jeux de données ainsi que la génération et la réutilisation des modèles, tout en vérifiant la préservation de l'information requise pour les analyses subséquentes. La deuxième expérience a évalué la réutilisation analytique du modèle au moyen d'une étude de cas de risque fondée sur la localisation, dans laquelle des indicateurs de risque météorologique et de risque de collision en vol (MAC) ont été dérivés indépendamment puis combinés. La troisième expérience a évalué l'identification et l'interprétation, fondées sur les données, de structures opérationnelles récurrentes à partir de trajectoires extraites du modèle d'espace aérien, au moyen d'une implémentation KMeans sélectionnée, d'une analyse de la granularité des catégories et d'une comparaison avec des données indépendantes de référence sur les aéronefs.

Les résultats montrent que des observations hétérogènes relatives aux aéronefs et aux conditions météorologiques peuvent être intégrées dans des sorties de modèles 4D réutilisables et indexées par DGGS tout en conservant l'information requise pour les analyses subséquentes. La sélection de variables a réduit le temps d'exécution des modèles et la taille des sorties enregistrées; le partitionnement des jeux de données a considérablement réduit les besoins en mémoire au prix d'un temps de traitement supplémentaire; et la réutilisation de sorties de modèles générées antérieurement a permis des analyses répétées sensiblement plus rapides. L'expérience portant sur le risque a démontré que des indicateurs dérivés indépendamment des données d'aéronefs et des données météorologiques peuvent être co-indexés, interprétés et combinés selon les mêmes références spatiales et temporelles. L'expérience de regroupement a permis d'identifier des structures de trajectoires récurrentes en environnement terminal et à plus haute altitude sans recourir à des étiquettes comportementales prédéfinies. L'analyse spatiale et cinématique effectuée après le regroupement non supervisé a permis d'interpréter ces structures comme des catégories de comportement opérationnel; la comparaison des solutions $K = 5$ et $K = 9$ a démontré une granularité de catégorisation ajustable; et la comparaison avec des données indépendantes de référence sur les aéronefs a montré que des catégories comportementales opérationnellement distinctes peuvent partager la même classification conventionnelle dominante des aéronefs.

Dans leur ensemble, les trois expériences démontrent un cadre réutilisable de modélisation de l'espace aérien fondé sur les données qui sépare la représentation persistante de données historiques hétérogènes de

l’espace aérien des analyses propres aux différentes applications qui lui sont ensuite appliquées. La méthodologie permet une génération et une réutilisation des modèles réalisables sur le plan computationnel, la dérivation et la combinaison de multiples produits analytiques fondés sur la localisation, l’identification fondée sur les données et l’interprétation postérieure au regroupement de structures opérationnelles récurrentes, ainsi que l’analyse d’environnements complexes d’espace aérien 4D appuyée par la visualisation.

Statement of contributions

This thesis makes four contributions to new scientific knowledge and engineering methodology in airspace modelling and aerospace data analysis.

1. Location-based formulation of a modular, reusable DGGS-based 4D airspace model

The airspace-modelling literature reviewed in this thesis contains trajectory-, region-, and location-based approaches developed for a wide range of operational and analytical purposes, with trajectory-based approaches making up the majority of existing models. However, the identified implementations are commonly specialized according to particular aircraft or data types, spatial representations, geographical scales, dimensional representations, or intended analytical applications. Their underlying airspace representations are therefore often closely coupled to the assumptions and analytical purposes for which they were developed, limiting their modular reuse when these requirements change.

This thesis contributes *a different formulation of the airspace modelling problem* in which fixed three-dimensional airspace locations and time, rather than individual aircraft trajectories or predefined operational regions, form the common reference structure of the model. This formulation is implemented using a four-dimensional DGGS-based representation in which heterogeneous aircraft and weather observations are associated with common indexed spatial cells and time references. The resulting model separates the persistent representation of the airspace from the analytical methods subsequently applied to it, providing a modular, reusable, multi-source airspace representation that is not constructed around one predefined analytical application.

2. Derivation of multiple location-based analytical products from a reusable 4D airspace model

Existing airspace models produce analytical outputs including trajectory predictions, congestion estimates, conflict alerts, weather products, and risk indicators, but the reviewed models are commonly developed around the particular analytical output they are intended to generate. This closely couples the underlying airspace representation to a predefined analytical purpose. Location-based models demonstrate that individual analytical quantities can be associated with spatial locations, but reuse of the same heterogeneous 4D representation to generate multiple derived analytical products at common locations and times is less established.

This thesis contributes *a methodology for deriving co-indexed analytical products from the contents of a reusable 4D airspace model without reconstructing the underlying airspace representation for each application*. As a specific realization of this methodology, a location-based airspace risk formulation is developed in which aircraft-proximity risk and hazardous-weather risk are independently derived from heterogeneous historical data at common three-dimensional DGGS locations and time steps and subsequently combined into a multi-hazard analytical field. The case study demonstrates that the developed airspace model can serve not only as a common structure for organizing historical data, but also as an analytical foundation from which new spatially and temporally referenced information can be generated.

3. Data-driven behaviour-based aircraft categorization using unsupervised learning and post-clustering interpretation

Historical trajectory data contain recurring spatial, temporal, and operational structure that can provide information about how aircraft operate within an airspace. However, identifying and interpreting this structure is challenging when the relevant operational groupings are not known in advance. Existing analytical approaches can require categories, labels, rules, or analytical boundaries to be defined through subject-matter expertise or application-specific assumptions before the underlying structure of the data has been examined.

This thesis contributes a *behaviour-based aircraft categorization approach* in which trajectory-level outputs from the reusable DGGS-based 4D airspace model are first grouped using unsupervised learning without predefined behavioural labels and are subsequently interpreted using spatial, kinematic, and domain evidence. Complete aircraft trajectories are treated as the objects being categorized, allowing recurring operational structure to be derived from their observed spatial and temporal evolution. Aircraft type, airworthiness category, runway information, and other domain knowledge are introduced only after clustering to support interpretation and independent comparison rather than to define the initial groups. Category granularity can be selected according to the intended analytical resolution while the membership and operational characteristics of the resulting groups remain derived from the trajectory data. The resulting categorization therefore describes how aircraft operate within the observed airspace and provides information that is distinct from and complementary to conventional technical aircraft classifications.

4. Computational evaluation of data-intensive airspace model generation and reuse

The literature identifies feature selection, dataset partitioning, parallelization, indexing, and other data-management strategies for improving computational efficiency. However, the computational behaviour of location-based, data-driven 4D airspace models remains insufficiently characterized under controlled conditions in which efficiency strategies are evaluated while preserving the model information required for downstream analyses. Without this distinction, apparent computational improvements can result from reducing or altering the analytical task rather than processing the required information more efficiently.

This thesis contributes *a methodology for evaluating the computational behaviour of airspace-model generation and reuse while holding the underlying analytical task and model requirements constant*. Feature selection, dataset partitioning, and model BUILD and LOAD strategies are evaluated across increasing historical data volumes using runtime, memory demand, saved model-output size, and model-reuse performance. This provides a reproducible means of distinguishing computational improvements resulting from more efficient processing from apparent improvements obtained by reducing or changing the information retained by the airspace model.

Table 1: Supporting Journal and Conference Publications

Title of paper	An Aircraft Categorization Approach using Machine Learning to Analyze Aircraft Behaviour
Authors	Nicolas Vincent-Boulay, Catharine Marsden
Type	Journal article
Journal/Conference	Journal of Air Transportation
Status	Published
Reference number	[1]
URL Link	https://arc-aiaa-org.journal.rmc.ca/doi/abs/10.2514/1.D0398
Title of paper	A geospatial approach to modelling airspace risk factors
Authors	Nicolas Vincent-Boulay, Catharine Marsden
Type	Journal article
Journal/Conference	Journal of Open Aviation Science
Status	Published
Reference number	[2]
URL Link	https://journals.open.tudelft.nl/joas/article/view/7402
Title of paper	A comparative evaluation methodology of unsupervised clustering algorithms for ADS-B trajectory-based aircraft behaviour analysis
Authors	Nicolas Vincent-Boulay, Catharine Marsden
Type	Journal article
Journal/Conference	Computing and Artificial Intelligence
Status	Accepted for publication on August 11, 2026
Reference number	[3]
URL Link	N/A
Title of paper	A new approach to aircraft categorization using machine learning to analyze aircraft behaviour
Authors	Nicolas Vincent-Boulay, Catharine Marsden
Type	Conference paper
Journal/Conference	AIAA Aviation 2023
Status	Published in proceedings & presented at conference
Reference number	[4]
URL Link	https://arc-aiaa-org.journal.rmc.ca/doi/abs/10.2514/6.2023-4214
Title of paper	Spatial modelling of midair collision risk using ADS-B data
Authors	Nicolas Vincent-Boulay, Catharine Marsden
Type	Conference paper
Journal/Conference	The 11th OpenSky Symposium
Status	Published in proceedings & presented at conference
Reference number	[5]
URL Link	https://journals.open.tudelft.nl/joas/article/view/7202
Title of paper	Data-driven Geospatial Modelling for Complex Airspace Environments
Authors	Nicolas Vincent-Boulay, Catharine Marsden, Angelina X. Cui
Type	Conference paper
Journal/Conference	AIAA SCITECH 2024 Forum
Status	Published in proceedings & presented at conference
Reference number	[6]
URL Link	https://doi-org.journal.rmc.ca/10.2514/6.2024-0571

Software Availability and Reproducibility

The integrated Python implementation developed for this thesis is maintained in the PhD Thesis Airspace Model GitHub repository. This repository contains the computational framework supporting all three research experiments, including the DGGS-based 4D airspace model, location-based risk analysis, unsupervised machine learning, and associated visualization and evaluation utilities.

Three complementary public repositories provide research implementations associated with the publications listed above: Geospatial Airspace Risk Model, Trajectory Clustering Comparison Methodology, and Aircraft Behaviour Categorization Model.

The repositories contain source code and supporting documentation describing their respective implementations and reproducibility requirements. The public companion repositories include self-contained demonstrations but do not redistribute the complete historical aerospace datasets or reproduce all reported experimental results independently. Reproduction of the full thesis experiments requires the corresponding datasets, experimental configurations, and computational environment.

Table of contents

List of tables xiii
List of figures xiv
List of abbreviations and acronyms xviii
List of symbols xx
Chapter 1: Introduction 1
1.1 Big data 1
1.2 Big data in aerospace 3
1.3 Airspace modelling 6
1.4 Changes affecting the airspace environment 8
1.5 Main research objective 9
1.6 Thesis organization 10
Chapter 2: Literature review 11
2.1 Background on airspace models 11
2.2 Airspace model literature review 29
2.3 Machine learning for airspace modelling 49
2.4 Synthesis of the literature review observations and research gaps 65
Chapter 3: Research approach 70
3.1 DGGS-based airspace model development and computational evaluation 72
3.2 Derivation of location-based airspace metrics 73
3.3 Behaviour-based aircraft categorization using machine learning 74
Chapter 4: Methodology 77
4.1 Airspace entity and data source selection 78
4.2 Data collection and preprocessing 78
4.3 Computational execution framework 80
4.4 DGGS-based 4D airspace model construction 80
4.5 Location-based airspace metric computation 82
4.6 Unsupervised machine learning framework 82
4.7 Visualization and interpretation tools 83
4.8 Methodology implementation 83
Chapter 5: DGGS-based airspace model development and evaluation 85

5.1 Experimental design, inputs, and outputs 85
5.2 Airspace model implementation 95
5.3 Airspace model development results and visualization 111
5.4 Computational evaluation results 124
5.5 Experiment #1 discussion 131
Chapter 6: DGGS-based airspace risk analysis 133
6.1 Experimental design, inputs, and outputs 134
6.2 Risk metric formulation 135
6.3 Experimental scenarios 143
6.4 Location-based risk metric results 144
6.5 Experiment #2 discussion 169
Chapter 7: Behaviour-based aircraft categorization using machine learning 174
7.1 Experimental objective and workflow 175
7.2 Trajectory dataset and clustering representation 177
7.3 Behaviour-based aircraft categorization results 179
7.4 Effect of category granularity 186
7.5 Validation through comparison with independent aircraft-reference categories 189
7.6 Experiment #3 discussion 191
Chapter 8: Discussion 196
8.1 Response to the research questions and hypotheses 197
8.2 Cross-experiment synthesis 200
8.3 Scope and limitations of the research 201
Chapter 9: Conclusion 205
9.1 Main conclusions 205
9.2 Future work 206
9.3 Final remarks 207
References 208
Appendix A – DGGS framework theoretical background used to support the methodology 236
A.1 DGGS base grid shape selection 236
A.2 Base polyhedron definition 238

A.3 Projection method 239
A.4 Grid subdivision 240
A.5 Data mapping and aggregation 240
A.6 Cell indexing 241
Appendix B – Supporting literature review for location-based airspace risk analysis 243
B.1 Purpose and scope 243
B.2 MAC risk models 243
B.3 Weather-related operational risk models 244
B.4 Requirements identified for the Experiment #2 risk model 244
B.5 Summary and implication for Experiment #2 245
Appendix C – Supporting clustering algorithm comparison and KMeans selection 246
C.1 Dataset characterization 246
C.2 Candidate clustering algorithm comparison 247
C.3 Selection of KMeans for the aircraft categorization application 261

List of tables

Table 1: Supporting Journal and Conference Publications viii
Table 2: The different applications for big data in aerospace [28] 4
Table 3: Airspace models 12
Table 4: Key differences between individual and aggregate predictive trajectory models 16
Table 5: CD&R models classified based on operational phases 17
Table 6: CD&R models classified based on modelling approach 18
Table 7: Summary of the different system-wide airspace simulation models 21
Table 8: Feature comparison of the three spatial modelling paradigms 30
Table 9: Airspace model comparison with respect to each identified observation 48
Table 10: Summary of the clustering algorithms 64
Table 11: Summary of the main Python files used in the implementation of the methodology 84
Table 12: Summary of inputs and outputs to the airspace model development and evaluation experiment 86
Table 13: Airspace model properties used in Experiment #1 87
Table 14: Historical time-range and dataset characteristics 91
Table 15: The different scenarios used to evaluate computational efficiency in Experiment #1 92
Table 16: Summary of the variables used for each scenario for testing the impact of FS 94
Table 17: Role of each data source used in Experiment #1 95
Table 18: List of available parameters in the OpenSky Network's dataset [450] 99
Table 19: List of available parameters in the NOAA dataset [463] 108
Table 20: Summary of the results and their corresponding validation requirement 124
Table 21: Inputs, outputs, and validation purpose of Experiment #2 components 135
Table 22: List of variables used in Experiment #2 and their descriptions 140
Table 23: Description of the different scenarios used in Experiment #2 143
Table 24: Validation of Experiment #2 results against the location-based risk model requirements 171
Table 25: Validation of Experiment #3 results against the success criteria 194
Table 26: Summary of the experimental scope of the thesis 202
Table 27: Summary of the principal limitations of the research and corresponding validation needs 204
Table 28: Comparison of DGGS basic shapes and their properties [490], [491] 237
Table 29: Risk model requirements identified for Experiment #2 245
Table 30: Clustering algorithms tested in Experiment #3 248
Table 31: Description of the hyperparameters varied for each candidate clustering algorithm 249
Table 32: Summary of the candidate algorithm comparison 260

List of figures

Figure 1: The “3 V’s” of big data [11] 1
Figure 2: Number of publications mentioning “big data” on the Web of Science database [27] 2
Figure 3: Number of publications mentioning “big data” in the field of aerospace engineering [27] 3
Figure 4: Key components of congestion prediction models 13
Figure 5: Key components of predictive trajectory models 15
Figure 6: Key components of CD&R models 17
Figure 7: Key components of flow control models 18
Figure 8: Key components of system-wide airspace simulation models 21
Figure 9: Key components of geospatial airspace models 23
Figure 10: Key components of unconventional aircraft airspace models 26
Figure 11: Key components of airspace weather models 27
Figure 12: Depictions of the three spatial modelling paradigms 30
Figure 13: Types of machine learning 49
Figure 14: Supervised and unsupervised machine learning algorithms 50
Figure 15: Unsupervised ML clustering algorithms 56
Figure 16: Depictions of the three modelling paradigms 66
Figure 17: Research approach 71
Figure 18: Overview of the methodology 77
Figure 19: Relationship between methodology and research experiments 78
Figure 20: Main steps for data collection and processing 79
Figure 21: Functional workflow diagram of the Python code 84
Figure 22: 3D DGGS structure of Airspace model #1 88
Figure 23: 2D view of Airspace model #1 showing LaGuardia airport 89
Figure 24: 3D DGGS structure of Airspace model #2 90
Figure 25: 2D view of Airspace model #2 showing major airport locations 90
Figure 26: Overview of the airspace model methodology steps implemented in Experiment #1 95
Figure 27: Air traffic data collection and processing 96
Figure 28: OpenSky Network ADS-B data coverage [452] 97
Figure 29: The process for accessing data using Trino 98
Figure 30: Local dataset storage structure 103
Figure 31: Example of a processed database file 104
Figure 32: Weather data collection and processing 105
Figure 33: NOAA NEXRAD weather radar data coverage [458] 106
Figure 34: KOKX weather radar station location (yellow square) and coverage (green region) [458] 107
Figure 35: The process for importing weather data from Amazon S3 servers 108

Figure 36: 3D visuals of the sequential processing steps of weather data from raw reflectivity values to storm cell centroids 109
Figure 37: 2D map of air traffic data for Airspace model #1 112
Figure 38: 3D map of air traffic data for Airspace model #1 114
Figure 39: 2D map of weather data for Airspace model #1 115
Figure 40: 3D map of weather data for Airspace model #1 116
Figure 41: 2D map of air traffic data for Airspace model #2 118
Figure 42: 3D map of air traffic data for Airspace model #2 120
Figure 43: 2D map of weather data for Airspace model #2 121
Figure 44: 3D map of weather data for Airspace model #2 122
Figure 45: Composite maximum reflectivity aggregation for Airspace model #2 123
Figure 46: BUILD runtime across increasing workloads 125
Figure 47: Mean RSS scaling 127
Figure 48: Peak RSS scaling 127
Figure 49: Effect of feature selection on saved model-output size 128
Figure 50: Cost of building versus reusing the baseline airspace model 129
Figure 51: Model reuse time across scenarios 130
Figure 52: Relationship between methodology and Experiment #2 133
Figure 53: Illustration of the shortest distances at two different times 138
Figure 54: Pseudocode used to calculate the weather risk metric 140
Figure 55: Pseudocode used to calculate the MAC risk metric 142
Figure 56: Airspace model used to derive risk metric results 144
Figure 57: High-level storm cell context for the weather risk scenario 145
Figure 58: 5km region near LGA airport with centroids 1 and 2 and storm data 146
Figure 59: Weather risk metric over time for centroids 1 and 2 147
Figure 60: 3D normalized weather risk indicator field at 2022-11-22 18:02:08 148
Figure 61: 3D normalized weather risk indicator field at 2022-11-22 18:03:58 149
Figure 62: 3D normalized weather risk indicator field at 2022-11-22 18:07:58 150
Figure 63: 2D weather risk indicator field at 1,200 m altitude for three selected times 151
Figure 64: 2D weather risk indicator field at 300 m altitude for three selected times 151
Figure 65: 5km region near LGA airport with centroids 3 and 4 and air traffic data 152
Figure 66: MAC risk metric over time for centroids 3 and 4 153
Figure 67: 3D normalized MAC risk indicator field for the two-aircraft scenario at 2022-11-22 18:06:05 154
Figure 68: 3D normalized MAC risk indicator field for the two-aircraft scenario at 2022-11-22 18:06:29 155
Figure 69: 3D normalized MAC risk indicator field for the two-aircraft scenario at 2022-11-22 18:06:42 156
Figure 70: 2D normalized MAC risk indicator field at 500 m for the two-aircraft scenario at three selected times 157

Figure 71: 2D normalized MAC risk indicator field at 2,400 m for the two-aircraft scenario at three selected times 157
Figure 72: 3D normalized MAC risk indicator field for the three-aircraft scenario at 2022-11-22 18:07:24 158
Figure 73: 3D normalized MAC risk indicator field for the three-aircraft scenario at 2022-11-22 18:07:32 159
Figure 74: 3D normalized MAC risk indicator field for the three-aircraft scenario at 2022-11-22 18:07:40 160
Figure 75: 2D normalized MAC risk indicator field at 500 m for the three-aircraft scenario at three selected times 161
Figure 76: 2D normalized MAC risk indicator field at 2,400 m for the three-aircraft scenario at three selected times 161
Figure 77: Centroids 1, 2, 3, and 4 with storm and air traffic data 162
Figure 78: Combined risk metric over time for centroids 1, 2, 3, and 4 163
Figure 79: 3D normalized combined weather–MAC risk indicator field at 2022-11-22 18:06:00 164
Figure 80: 3D normalized combined weather–MAC risk indicator field at 2022-11-22 18:06:06 165
Figure 81: 3D normalized combined weather–MAC risk indicator field at 2022-11-22 18:06:54 166
Figure 82: 2D combined weather–MAC risk indicator field at 1,200 m at three selected times 167
Figure 83: 2D combined weather–MAC risk indicator field at 300 m at three selected times 167
Figure 84: 2D combined weather–MAC risk indicator field at 500 m at three selected times 168
Figure 85: 2D combined weather–MAC risk indicator field at 2,400 m at three selected times 168
Figure 86: Relationship between methodology and Experiment #3 174
Figure 87: Experiment #3 workflow for data-driven identification and interpretation of aircraft behaviour categories 176
Figure 88: 2D view of the trajectory-level clustering input dataset 177
Figure 89: 3D view of the trajectory-level clustering input dataset 178
Figure 90: Aligned parameters over normalized time 179
Figure 91: 2D view of the K=5 KMeans trajectory clusters 180
Figure 92: Altitude over normalized time for each trajectory coloured by cluster ID 181
Figure 93: Cluster-size distribution 181
Figure 94: Heading profiles over normalized trajectory time 182
Figure 95: Vertical rate over normalized time 183
Figure 96: Velocity over normalized time 184
Figure 97: Normalized aircraft behaviour variable heatmap for the K=5 KMeans model 185
Figure 98: Cluster distribution comparison for the k=5 and k=9 KMeans models 186
Figure 99: 2D map of clustered trajectories for a) k=5 and b) k=9 KMeans models 187
Figure 100: Altitude over time comparison for the k=5 and k=9 KMeans models 188
Figure 101: Vertical rate over time comparison for the k=5 and k=9 KMeans models 189
Figure 102: Airworthiness category distribution of the input ADS-B dataset 190

Figure 103: Within-cluster distribution of independent aircraft-reference categories for the K=5 KMeans trajectory groups 191
Figure 104: Research gaps addressed by the thesis 196
Figure 105: Overview of the validation structure of the thesis 197
Figure 106: Main steps for creating the DGGS framework 236
Figure 107: Comparison of different base polyhedrons (taken from [490], [491]) 239
Figure 108: Visual representation of 1-resolution level Morton codes in a 3D space for a cubic grid (taken from [497]) 241
Figure 109: Histograms of the x, y and z(altitude) data distributions 247
Figure 110: Boxplots of the x, y and z(altitude) data distributions 247
Figure 111: Comparison of Silhouette coefficient values for the 5 shortlisted candidates from each clustering algorithm 251
Figure 112: Noise fractions for the shortlisted DBSCAN and HDBSCAN candidate models 252
Figure 113: Comparison of Calinski–Harabasz values for the 5 shortlisted candidates from each clustering algorithm 253
Figure 114: Comparison of Davies–Bouldin values for the 5 shortlisted candidates from each clustering algorithm 254
Figure 115: Effective cluster count relative to the five exploratory behaviour families identified in the input dataset 255
Figure 116: 2D trajectory comparison for the representative DBSCAN, HDBSCAN, KMeans, and GMM configurations 257
Figure 117: 3D trajectory comparison for the representative DBSCAN, HDBSCAN, KMeans, and GMM configurations 258
Figure 118: Aligned altitude over time comparison for the representative DBSCAN, HDBSCAN, KMeans, and GMM configurations 260

List of abbreviations and acronyms

Acronym	Description
AAM	Advanced Air Mobility
ADS-B	Automatic Dependent Surveillance–Broadcast
AI	Artificial Intelligence
ANSP	Air Navigation Service Provider
API	Application Programming Interface
ARTCC	Air Route Traffic Control Center
ATC	Air Traffic Control
ATFM	Air Traffic Flow Management
ATM	Air Traffic Management
AWS	Amazon Web Services
BIRCH	Balanced Iterative Reducing and Clustering using Hierarchies
CD&R	Conflict Detection and Resolution
CFD	Computational Fluid Dynamics
CNN	Convolutional Neural Network
CRM	Collision Risk Model
CSV	Comma-Separated Values
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
DGGS	Discrete Global Grid System
DL	Deep Learning
DP	Dataset Partitioning
EWR	Newark Liberty International Airport
FAA	Federal Aviation Administration
FCM	Fuzzy C-Means
FDR	Flight Data Recorder
FS	Feature Selection
GIS	Geographic Information System
GMM	Gaussian Mixture Model
GPS	Global Positioning System
HDBSCAN	Hierarchical Density-Based Spatial Clustering of Applications with Noise
ICAO	International Civil Aviation Organization
IoT	Internet of Things
JFK	John F. Kennedy International Airport
LGA	LaGuardia Airport

LOF	Local Outlier Factor
LSTM	Long Short-Term Memory
MAC	Midair Collision
ML	Machine Learning
NAS	National Airspace System
NASA	National Aeronautics and Space Administration
NCEI	National Centers for Environmental Information
NEXRAD	Next Generation Radar
NOAA	National Oceanic and Atmospheric Administration
PCA	Principal Component Analysis
Py-ART	Python ARM Radar Toolkit
RL	Reinforcement Learning
RSS	Resident Set Size
SG	Savitzky–Golay
SQL	Structured Query Language
sUAS	Small Unmanned Aircraft System
SVM	Support Vector Machine
TEB	Teterboro Airport
TFM	Traffic Flow Management
t-SNE	t-Distributed Stochastic Neighbor Embedding
UAM	Urban Air Mobility
UAS	Unmanned Aircraft System
UAV	Unmanned Aerial Vehicle
UTC	Coordinated Universal Time
UTM	Universal Transverse Mercator
WCSS	Within-Cluster Sum of Squares
WGS84	World Geodetic System 1984
XGBoost	Extreme Gradient Boosting

List of symbols

Symbol	Description
a	Semi-major axis of the WGS84 reference ellipsoid
A	Auxiliary term used in the UTM projection equations
a_k	Aircraft k in the airspace region of interest
AP_m	Aircraft pair m
c_j	DGGS centroid j
c_k	Savitzky–Golay convolution coefficient at index k
C	Auxiliary term used in the UTM projection equations
CellID	Identifier assigned to a DGGS cell
CR_{c_j,t_i}	Combined weather-MAC risk indicator at centroid c_j at time t_i
CT_{a_k,c_j,t_i}	Convergence time of aircraft a_k toward centroid c_j at time t_i
CT_{s_k,c_j,t_i}	Convergence time of storm cell s_k toward centroid c_j at time t_i
CT_{AP_{m1},c_j,t_i}	Convergence time of the first aircraft in aircraft pair AP_m
CT_{AP_{m2},c_j,t_i}	Convergence time of the second aircraft in aircraft pair AP_m
$CT_{\min,AP_m,c_j,t_i}$	Minimum convergence time of aircraft pair AP_m
d_{a_k,c_j,t_i}	Shortest distance between aircraft a_k and centroid c_j at time t_i
d_{s_k,c_j,t_i}	Shortest distance between storm cell s_k and centroid c_j at time t_i
e^2	Squared eccentricity of the WGS84 reference ellipsoid
E	UTM Easting coordinate
E_0	Easting offset used in the local planar-coordinate formulation in Appendix A
f	Flattening parameter of the WGS84 reference ellipsoid
k_0	UTM scale factor at the central meridian
m	Half-window size of the Savitzky–Golay filter
M	Meridional arc used in the UTM projection equations
$MACR_{AP_m,c_j,t_i}$	MAC risk metric for aircraft pair AP_m, centroid c_j, and time t_i
$MACR_{c_j,t_i}$	Location-based MAC risk metric at centroid c_j and time t_i
$MACR^N_{c_j,t_i}$	Normalized location-based MAC risk metric at centroid c_j and time t_i
N	Radius-of-curvature term used in the UTM projection equations
n_{level}	Number of cells associated with a DGGS subdivision level in Appendix A
p	Polynomial order of the Savitzky–Golay filter
s_k	Storm cell k in the airspace region of interest
t_i	Current analysis time
t_{i-1}	Previous analysis time

T	Auxiliary term used in the UTM projection equations
v_{a_k,c_j,t_i}	Convergence velocity of aircraft a_k toward centroid c_j
v_{s_k,c_j,t_i}	Convergence velocity of storm cell s_k toward centroid c_j
W	Savitzky–Golay filter window size
WR_{c_j,t_i}	Weather risk metric at centroid c_j and time t_i
$WR^N_{c_j,t_i}$	Normalized weather risk metric at centroid c_j and time t_i
X	UTM Easting / Cartesian x-coordinate used by the airspace model
x	Local planar x-coordinate used in the DGGS formulation
Y	UTM Northing / Cartesian y-coordinate used by the airspace model
y	Local planar y-coordinate used in the DGGS formulation
$y[n]$	Original signal value at sample n in the Savitzky–Golay formulation
$y'[n]$	Filtered signal value at sample n
$Z,\ z$	Vertical Cartesian coordinate / geometric altitude
$\Delta CT_{AP_m,c_j,t_i}$	Difference between the two convergence times of aircraft pair AP_m
$\Delta CT_{\text{threshold}}$	Maximum permitted aircraft-pair convergence-time difference
$\Delta d_{a_k,c_j,t_i}$	Change in aircraft-to-centroid distance between consecutive time steps
$\Delta d_{s_k,c_j,t_i}$	Change in storm cell-to-centroid distance between consecutive time steps
Δt	Time interval between t_i and t_{i-1}
Δx	DGGS cell dimension in the x-direction
Δy	DGGS cell dimension in the y-direction
ϕ	Latitude in radians
λ	Longitude in radians
λ_0	Longitude of the central meridian of the UTM zone

Chapter 1: Introduction

This chapter introduces fundamental concepts in big data, airspace modelling, and machine learning. It highlights the importance of leveraging big aerospace data for accurate modelling of real-world airspace operations and underscores the challenges and constraints associated with using machine learning for airspace modelling. It explores the advantages of adopting a spatial rather than an aircraft-based approach to airspace modelling and discusses the role of computationally efficient algorithms and processes in big data. The chapter concludes with a summary of the dissertation's main focus.

1.1 Big data

The term 'big data' refers to datasets that are so voluminous and complex that traditional data processing software is inadequate to deal with them [7]. Big data is often characterized by the "3 V's" [23–25]: Volume, Velocity, and Variety. Volume refers to the quantity of generated and stored data. Velocity refers to the rapid rate at which new data is produced (and sometimes used); and Variety refers to the different types and nature of the data such as its structure, its provenance, or the type of information it captures.

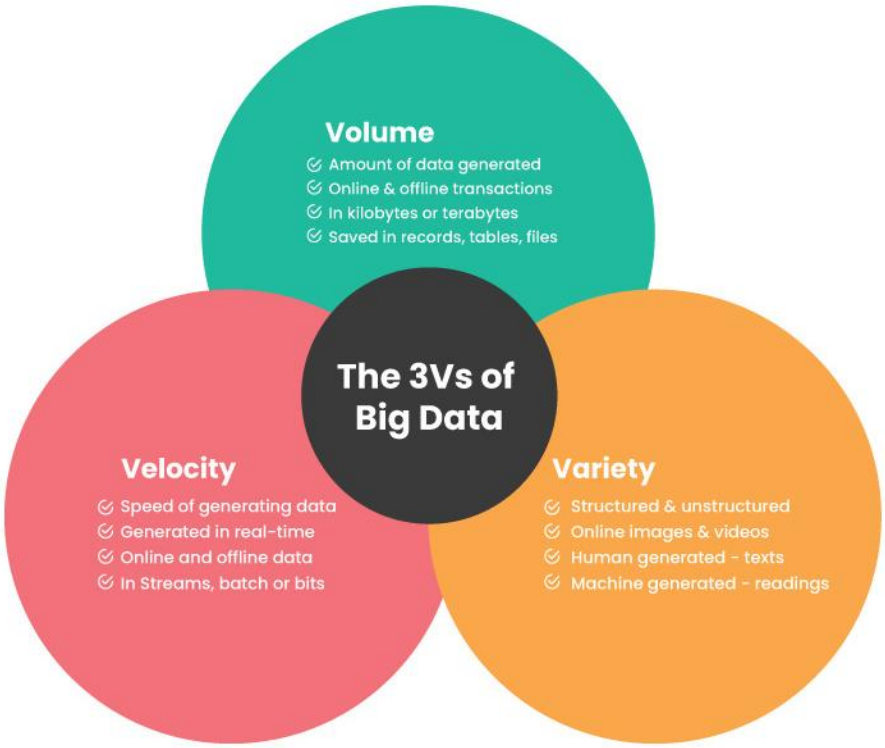


Figure 1: The "3 V's" of big data [11]

The past two decades have been characterized by an exponential growth in the amount of data being generated across many fields [12]. The origin of this growth can be traced back to the advent of the internet and the digital revolution of the early 2000s [13]. Digital data was initially generated through desktop computing and early web activities, but the amount has increased dramatically since the introduction of smartphones, social media, and Internet of Things (IoT) devices [13]. In 2017, Domo [14], a business intelligence software company, estimated that 90% of all the existing data in the world was created in the last two years (2015-2016), equivalent to approximately 2.5 quintillion bytes of new data per day. In 2024,

they estimated the total amount of data created, captured, copied, and consumed globally to be 149 zettabytes (one zettabyte is a trillion gigabytes), with projections surpassing 394 zettabytes by 2028.

In healthcare, the use of big data enhances patient outcomes through predictive analytics for disease prevention, personalized medicine, and operational efficiencies in hospitals [15]. Predictive analytics can identify patient populations at risk for specific conditions, enabling early intervention and personalized treatment plans [15]. The financial and banking sectors utilize big data for fraud detection, risk management, and personalized banking services. Real-time analytics assist in monitoring transactions and identifying anomalous activities that may indicate fraudulent behaviour [16]. In marketing and retail, big data is used to analyze consumer behaviour, develop targeted advertising, improve demand forecasting, and optimize customer segmentation to enhance engagement and increase sales [17]. Governments leverage big data for public-sector applications, including policymaking, urban planning, fraud detection and smart governance; thereby enhancing efficiency, transparency, and responsiveness to societal needs [18]. One example of this is Singapore's Smart Nation initiative [18], which uses real-time data from sensors, Internet of Things (IoT) devices, and public services to optimize city operations. In agriculture, big data plays a role in optimizing crop yields, managing resources, and monitoring climate conditions, supporting sustainable farming practices and food security [19]. The education sector benefits from big data analytics by using it to facilitate personalized learning [20], track student performance [21], and improve institutional decision-making [22]. In transportation and logistics, big data supports traffic management, fleet optimization and predictive maintenance, reducing costs and improving efficiency in supply chains and mobility networks [23]. The energy sector uses big data in smart power grids, energy consumption forecasting and predictive maintenance, leading to more sustainable and efficient energy use [24]. In manufacturing, big data is used to enhance process optimization, defect detection, and supply chain management, allowing industries to improve productivity and reduce waste [25]. The cybersecurity sector also benefits from big data analytics, utilizing anomaly detection, intrusion prevention, and identity verification to mitigate threats and enhance digital security [26]. Figure 2 shows the annual number of publications returned by a Web of Science Core Collection Topic search for the phrase "big data." The results show that publications explicitly addressing big data began to increase rapidly in the early 2010s and remained at substantially higher levels throughout the following decade.

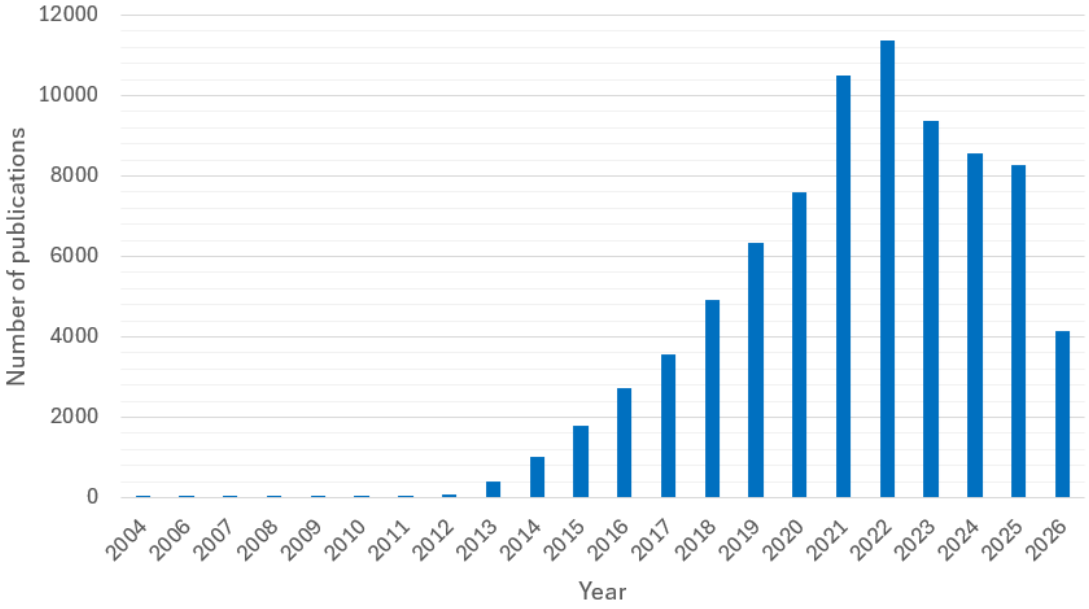


Figure 2: Number of publications mentioning "big data" on the Web of Science database [27]

1.2 Big data in aerospace

Big data is also leveraged in the aerospace sector. The use of big data in aerospace has evolved significantly over time, driven by advancements in data collection, storage, and analytical capabilities. In the early years of aviation (1900s to the 1940s), data usage was limited to manual record-keeping of flight operations and maintenance logs, with limited real-time insights [28]. The mid-20th century saw the introduction of radar and telemetry systems, enabling real-time monitoring of air traffic and aircraft performance [29]. By the 1980s and 1990s, the rise of digital avionics and flight data recorders (FDRs) allowed for more systematic data storage and analysis, paving the way for predictive maintenance and improved flight data monitoring [30]. The early 2000s marked a shift toward networked air traffic management systems, with the introduction of Automatic Dependent Surveillance–Broadcast (ADS-B), GPS, satellite-based and other real-time tracking technologies, improving situational awareness and airspace utilization [31]. The big data revolution of the 2010s accelerated aerospace data analytics, integrating cloud computing, artificial intelligence, and machine learning to process large amounts of sensor data from aircraft, satellites, and unmanned aerial vehicles (UAVs) [32].

More recently, the quantity of generated and stored data, the rapid rate at which new data is produced, and the variety of different types of aviation data (its structure, its provenance, the type of information it captures) are all increasing [33]. Figure 3 shows the annual number of publications returned by the same Web of Science Core Collection Topic search for "big data" when restricted to the Engineering, Aerospace category. Publications explicitly addressing big data begin to appear consistently within this category in the mid-2010s, followed by a general increase in research activity during subsequent years.

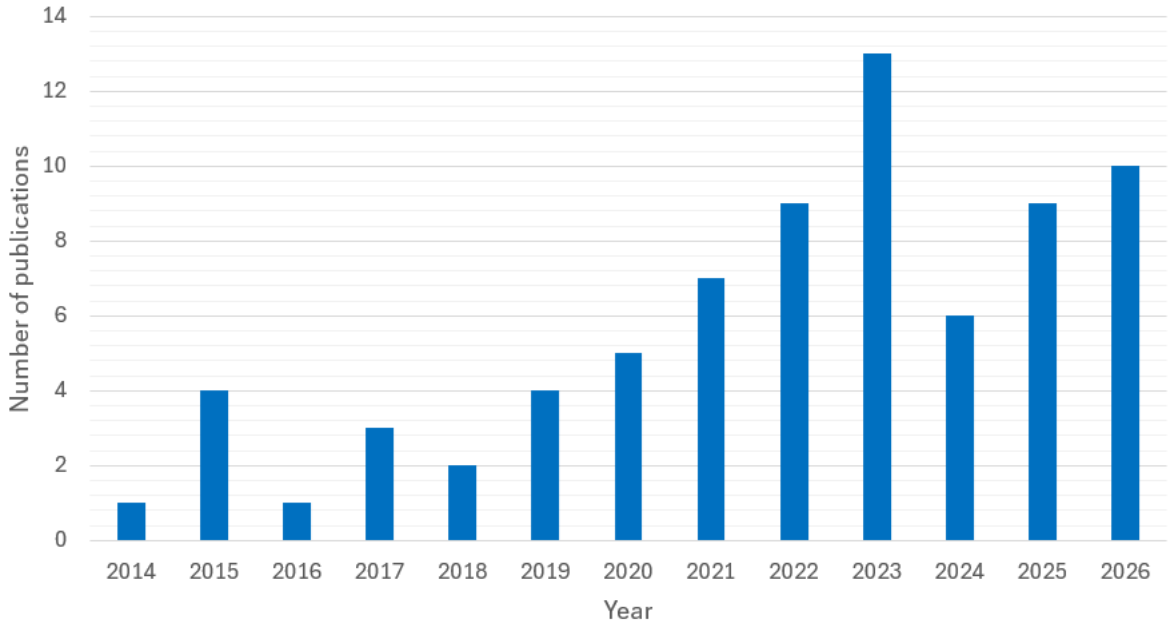


Figure 3: Number of publications mentioning "big data" in the field of aerospace engineering [27]

According to Adamopoulou and Daskalakis [28], applications for big data in aerospace can be categorized into six primary domains, as summarized in *Table 2*: aviation technology, aviation management, UAV-enabled applications, military aviation, health/environment-related applications, and space technology [28]. In aviation technology, big data is used to enhance aircraft design and manufacturing. In aircraft design it is used to enable real-time monitoring of flight parameters. In manufacturing it is used for improving

maintenance practices by predicting failures and reducing downtime. In aviation management, big data plays a major role in optimizing air traffic flow, trajectory prediction, delay mitigation, and resource allocation; all of which contribute to improving the efficiency of global airspace operations. The UAV sector benefits from big data for industrial monitoring, automated inventory tracking, surveillance, and disaster response. Military aviation relies on big data for characterizing and identifying aircraft, improving pilot safety, and optimizing mission strategies. In the environmental and health fields, big data facilitates the monitoring of aviation emissions, urban air quality, disease tracking, and epidemiological modelling; including its role in managing pandemics and mitigating health risks associated with air travel. Lastly, space technology leverages big data in supporting satellite operations, optimizing launch missions, processing spaceborne telemetry, and facilitating astronomical research through improved data management and analysis [28].

Chennu and Veeredhi [34] explain that the amount and diversity of data being produced and used by the aerospace industry is expected to continue increasing; and Bhatia et al. [35] predict that big data analytics is expected to play an increasingly critical role in ensuring safety, efficiency, and sustainability across most aerospace applications.

Table 2: The different applications for big data in aerospace [28]

Category	**Subcategory**
Big data in aviation technology	Aircraft design/manufacturing Monitoring of flight/aircraft/safety parameters Health monitoring systems
Big data in aviation management	Air traffic management and Trajectory planning Delay prediction and resource allocation Maintenance optimization Collecting customer insights/increasing Customer satisfaction
Big data and UAVs	Solutions for the industry Solutions for infrastructures
Big data in military aviation	Supporting military operations Increasing air force safety
Big data related to environmental and health aspects	Aviation big data related to Health aspects Aviation big data related to Environmental aspects
Big data in space technology	Supporting space missions Solutions utilizing spaceborne big data

Aerospace big data is characterized by several factors that influence its collection, processing, and application, which can be summarized as follows:

1. The *volume* of data generated in aerospace is typically very large, with sources such as satellite imagery, air traffic surveillance systems, and onboard aircraft sensors, which can produce petabytes (> 1,000,000 GB) of data daily according to estimates made by Lootens and Efthymiou [36].
2. The rate at which new data is produced and used in aerospace applications (sometimes referred as big data *velocity*) is very high. For example, Bach and Paielli [37] highlight that ground-based radar tracking used by air traffic controllers typically uses 12-second or 4.8-second sampling rates, while ADS-B tracking uses 1-second sampling, and smoothing radar data can provide 1-second tracking similar to ADS-B. Another example is Li et al. [38], who show that relative positioning in UAV and ship-based operations requires up to 125 Hz (0.008 seconds) data output rates.
3. Big aerospace data covers a wide range of applications and is thus characterized by high *variety* in the different types and nature of the data, such as its structure, its provenance, or the type of information it captures. For example, it includes spatial, temporal, audio, and other heterogeneous data types [28]. It also includes data that can be structured to varying degrees: structured data (flight logs, airspace configurations), semi-structured data (weather reports, maintenance records), and unstructured data (ATC voice recordings, remote sensing images) [39].

The volume, velocity, and variety characteristics discussed above are common to big data applications across different fields and are not unique to aerospace. What distinguishes aerospace big data is primarily the nature of the information involved and how these characteristics manifest, such as large volumes of aircraft surveillance and sensor data, high-frequency observations, and diverse aerospace data sources. Similarly, the need for efficient data management and processing is a general big data challenge, with the specific computational requirements depending on the characteristics of the datasets and their intended applications.

These three characteristics of aerospace big data lead to the first observation driving the thesis topic:

Observation #1

Aerospace big data is characterized by high volume, rapid velocity, and large variety.

Several studies highlight the necessity for big data applications to be efficient, particularly in managing large-scale datasets and optimizing computational resources.

For example, traditional data management systems are often inadequate for handling large volumes of big data, requiring the implementation of systems such as the Hadoop Distributed File System (HDFS) and parallel computing frameworks like MapReduce and Spark. Research by Singh et al. [40] discusses how these architectures enable efficient data storage and retrieval, addressing the challenge of large data volumes through optimized parallelization.

The rapid velocity at which data is generated presents further challenges, particularly in applications requiring real-time processing. Sindhu and Hegde [41] demonstrate that leveraging adaptive algorithms reduces latency and computational overhead while maintaining the integrity of high-speed data streams for datasets characterized by rapid velocity. Similarly, Bagga and Sharma [42] highlight the use of stream processing tools such as Apache Storm and Apache Flink to facilitate real-time analytics.

Beyond volume and velocity, the variety of big data further complicates its use, as datasets often contain structured, semi-structured, and unstructured formats originating from diverse sources. Arora [43] discusses the challenges posed by heterogeneous data structures and proposes a multi-representation approach for data integration, which enables the interaction between different data types. Another challenge associated with data variety is the need for adaptable query languages capable of handling complex and diverse datasets. Novikov et al. [44] explore the use of similarity-based searches and adaptive querying techniques to improve data retrieval from heterogeneous sources, ensuring that complex datasets can be processed effectively.

Research by Wu et al. [45] highlights that high-performance computing (HPC) and high computational efficiency are critical for managing big data applications which use data with high volume and rapid velocity, as traditional computing architectures struggle with increasing data complexity and processing demands. Similarly, Shu [46] explores divide-and-conquer methods, which involve breaking down large datasets into smaller, more manageable segments before analysis, showing the importance of computational efficiency when dealing with data with high volume and rapid velocity, while maintaining the accuracy of the dataset's use case.

The literature underscores the need for robust collection and processing techniques to effectively handle big data characterized by high volume, rapid velocity, and large variety, which can be summarized as:

Observation #2

Reviewed studies report that aerospace big data applications require data management and processing methods designed for effective handling of high-volume, rapidly updated, and heterogeneous datasets.

1.3 Airspace modelling

One of the ways to leverage big data in the aerospace field is to use it to create models of the airspace environment [47]. These computational representations are used to simulate and study interactions among air vehicles, their operations, and the broader air traffic ecosystem. The insights derived are used by stakeholders in a variety of aviation domains, including regulators, air navigation service providers (ANSPs), researchers, airlines, and manufacturers. Airspace modelling plays an important role in ensuring the safety, efficiency, and sustainability of aircraft operations, especially as the industry evolves and becomes increasingly complex [48], [49].

A variety of approaches exist for airspace modelling, each of them tailored to specific operational and analytical needs. These methods vary in scope, computational requirements, and focus, reflecting the diverse challenges of managing airspace effectively. *Congestion prediction models* predict airspace congestion by analyzing traffic density and flow patterns [50]. They often incorporate probabilistic approaches to forecast when and where congestion might occur, enabling proactive traffic management. *Predictive trajectory models* are another approach that focuses on forecasting the future trajectories of individual aircraft based on current flight paths, weather conditions, and other influencing factors [51]. *Conflict Detection and Resolution (CD&R) models* are designed to identify and resolve potential conflicts like midair collisions between aircraft in real time. These models evaluate the relative positions, velocities, and trajectories of aircraft to detect risks and recommend conflict resolution maneuvers such as altitude adjustments or rerouting [52]. *Flow control models* are a type of airspace model that are used to manage the flow of air traffic through specific airspace regions by optimizing route assignments, departure times,

and sequencing [53]. They aim to balance the workload across sectors and prevent overloading any area. Airspace modelling is also used to study aspects of the airspace other than aircraft movement, such as weather phenomena. *Airspace weather models* integrate real-time data (radar and satellite imagery) with meteorological simulations to forecast conditions including turbulence, wind, and storm patterns [54]. *Geospatial airspace models* focus on spatial analysis of the airspace using data from geographical information systems (GIS) and three-dimensional (3D) mapping [55]. These models analyze how the airspace is structured and utilized considering terrain, obstacles, and other physical constraints. *System-wide airspace simulation models* take a macroscopic view of the airspace system, simulating the interactions between various components including airports, air traffic control, and airlines. They are typically used for policy analysis and strategic planning (evaluating the impact of new regulations or technologies on airspace efficiency) [56]. *Unmanned Aerial Systems (UAS) airspace models* address the challenges posed by integrating new entrants, such as drones and electric Vertical Takeoff and Landing (eVTOL) vehicles, into airspace operations [57]. *Hybrid airspace models* often combine features from other approaches, such as geospatial and flow control models, to address specific needs for these vehicles [58].

Although the aerospace field has historically been dominated by the use of proprietary data, there has been a recent increase in the amount of open-source aerospace data [33]. For example, the OpenSky Network [59], a non-profit association based in Switzerland, is an open-source initiative that allows public access to Automatic Dependent Surveillance Broadcast (ADS-B) air traffic data. This initiative has enabled a range of research applications from traffic management to environmental monitoring [59]. Another example of emerging open data initiatives in aviation is OpenAP, an open-source aircraft performance model based on surveillance data and open literature models [60]. The model serves as a tool for simulation-based studies in air transportation, promoting openness and comparability in research.

The increase in the quantity and diversity of aerospace data (including open-source data) is an enabling factor for new data-driven research opportunities. A trend can be observed over the last 5 years where more airspace models are being developed or updated to make use of this data. For example, the MIT Lincoln Laboratory [61] developed a high-performance computing workflow for processing billions of aircraft observations from the OpenSky Network to train Bayesian Network models to capture statistically significant aircraft trajectories as a resource for studying the risk of midair collisions at low altitudes. This effort was aimed at supporting the integration of unmanned aircraft systems into the U.S. National Airspace System by quantifying the risk of airborne collisions between unmanned and manned aircraft. The same group developed a similar model [62] for terminal airspace operations, which was also trained using data from the OpenSky Network. Alligier [63] used historical ADS-B data and machine learning to predict aircraft mass and speed profiles during the climb phase. By predicting a Gaussian distribution for each parameter, the approach aimed to improve the accuracy of aircraft climb predictions, addressing the challenge of predicting aircraft trajectories with unknown mass and speed intent. In the field of aircraft performance modelling, Sun et al. [64] used ADS-B data to develop a method for estimating aircraft performance parameters (such as turn radius, bank angle, and load factor) during turns. Bolic et al. [65] explored trajectory clustering using ADS-B data to categorize air traffic, aiming to understand the common routing choices of airlines and how these choices relate to aircraft types and operational flight costs. The research revealed significant associations between trajectory clusters, airlines, aircraft types, and costs that could inform air traffic management and modelling.

Observation #3

In recent years, the development of data-driven airspace models has accelerated, partly driven by the growing availability of open-source aerospace big data.

1.4 Changes affecting the airspace environment

The airspace environment is undergoing two changes that are making it more difficult to model.

The first is that average yearly *air traffic volumes are rising* [66]. Between 1991 and 2016, the number of aircraft movements increased by approximately 140%, a trend that is expected to continue in the future, particularly in fast-growing regions such as Asia and the Middle East [67]. The rise in air traffic volume is not linear but rather exponential in some regions, leading to capacity constraints and greater temporal and spatial variability in air traffic flows [68]. The increasing air traffic demand is expected to continue exceeding airspace capacity, pushing Air Navigation Service Providers (ANSPs) to implement more sophisticated traffic management strategies, such as free route airspaces and performance-based navigation (PBN), to enhance efficiency [69].

Growing air traffic volumes are closely linked to *increasing volumes of air traffic data*, which play an important role in the development of airspace models, particularly as big data becomes more integrated into airspace management and research. Ayhan et al. [70] explain that with the rise in real-time data streams including aircraft trajectories, weather updates, and air traffic control instructions; the amount of data that must be processed and analyzed has increased significantly. This has prompted the investigation of new research avenues tailored to cope more effectively with large scale data in airspace models, such as machine learning-based models (discussed in the literature review of Chapter 2).

A second change that is affecting the airspace environment and making it more difficult to model is *the arrival and integration of new entrants*, such as UAS, eVTOL vehicles, and hybrid-electric aircraft. The integration of new types of aircraft into the airspace presents both opportunities and challenges in airspace modelling and research. For example, some of these new air vehicles provide innovative solutions for urban air mobility (UAM) and cargo transport, addressing capacity needs in large urban centers. However, they also introduce new and significant technical, regulatory, and operational challenges [71]. The growing diversity of new entrants includes not only different vehicle types but also vehicles with *varying capabilities, sizes, and operational requirements*. For example, drones and eVTOLs differ significantly from traditional aircraft (airplanes and rotorcraft) in terms of flight dynamics, altitude ranges, and communication needs [72]. Managing these differences, especially in urban airspaces where traffic is denser, poses significant issues related to safety, traffic separation, and infrastructure [73].

The *new types and diversity of operations* performed by these entrant air vehicles are also introducing questions in the field of airspace modelling. Traditional airspace has long supported a variety of operations, including long-distance airliner transport, regional airliner transport, helicopter-based operations and others. The rise of UAM and Advanced Air Mobility (AAM) has introduced new operational categories, such as air taxis, package delivery drones, and emergency medical transport services, which present significant differences when compared to traditional operations [74]. For example, Franciscone and Fernandes [73] highlight the need for real-time meteorological data integration and reliable air traffic control systems to ensure safe operations of eVTOLs in metropolitan regions, further complicating the expansion of these new operations. The introduction of new aircraft types and operations also necessitates

the development of air traffic management solutions that can handle the demands of high-density, low-altitude airspace while maintaining safety and efficiency [75].

These combined changes have the effect of increasing the overall complexity of the airspace environment and make it more difficult to model effectively. For example, necessitating multiple airspace models for different air vehicle types [76], or significantly slows down existing airspace models when integrating large datasets of air traffic data [70]. The National Academies of Sciences [77], [78], [79] explain that such changes to the airspace environment will continue to appear in the foreseeable future, and that unpredicted challenges must be expected. They suggest investigating research avenues which are based on data-driven and scalable approaches, such as machine learning models, and that can adapt to unexpected changes [79].

Observation #4

Air traffic volumes are increasing significantly, and this trend is expected to continue in the future.

Observation #5

New and unconventional types of air vehicles are currently being integrated into the airspace or are in the process of being integrated.

1.5 Main research objective

The approaches used to model the airspace and its constituent parts are each designed to address specific operational and analytical needs varying in scope, computational requirements, and focus. There has been a growing trend in the development of data-driven airspace models that leverage large-scale aerospace datasets, driven in part by the increasing availability of open-source data in a field historically dominated by proprietary sources.

As airspace complexity continues to grow, with increasing air traffic volumes and new unconventional aircraft entering the airspace, traditional modelling approaches must evolve to meet these demands and remain representative of real-world conditions.

Observations #1 through #5 lead to the main research objective for this thesis:

Main research objective

To develop a methodology for a data-driven airspace modelling approach that leverages big aerospace data using collection and processing techniques effective at managing large-scale datasets while optimizing computational resources.

1.6 Thesis organization

The remaining chapters of this thesis are structured as follows.

Chapter 2 provides a literature review of two major topics relevant to the main research objective: airspace models and machine learning. The review of the literature on airspace models focuses on data-driven airspace models that use big aerospace data and their effectiveness at managing large-scale datasets including large-scale air traffic data and new entrant air vehicles, while optimizing computational resources. The section on machine learning literature first introduces relevant concepts and then reviews their use in the field of airspace modelling. Chapter 2 concludes with a summary of the main observations made from the reviewed literature, which are used to identify the main research gaps that will be addressed in this thesis.

Chapter 3 outlines the research plan for the thesis, which is based on the observations and identified gaps in the previous chapters. It formulates the corresponding research questions and hypotheses. The chapter then describes the three research experiments used to test the hypotheses and validate the proposed methodology.

Chapter 4 describes the general methodology developed in this research. It presents the selection and preparation of airspace entities and data sources, the computational execution framework, and the construction of the airspace model. It also describes the two principal analytical branches supported by the model: the computation of location-based airspace metrics and the application of unsupervised machine learning. Visualization and interpretation tools, as well as the implementation of the methodology within the developed computational workflow, are also presented.

Chapter 5 presents Experiment #1, which develops and evaluates the proposed airspace model. Two representative model configurations are used to examine the methodology under different spatial, temporal, resolution, and data conditions. The resulting models are evaluated through visual inspection and computational performance measurements under alternative execution strategies.

Chapter 6 presents Experiment #2, which uses the developed airspace model to derive and interpret location-based analytical products through a multi-hazard risk case study. Weather risk, MAC risk, and combined weather–MAC indicators are computed at common DGGS locations and time steps and examined across space, altitude, and time. The experiment evaluates whether heterogeneous, time-varying information contained in the reusable 4D model can support coherent and interpretable downstream analytical products.

Chapter 7 presents Experiment #3, which evaluates whether trajectory-level outputs from the reusable 4D airspace model can support data-driven identification and interpretation of recurring operational patterns, using behaviour-based aircraft categorization as the selected case study. The selected KMeans implementation is applied to the trajectory dataset, and the resulting groups are examined through spatial and kinematic characterization, category-granularity analysis, and comparison with independent aircraft-reference information. The supporting clustering-algorithm comparison used to select KMeans is presented separately in Appendix C.

Chapter 8 provides an integrated discussion of the thesis results. It evaluates the three research questions and hypotheses, synthesizes the findings that emerge across the experiments, and defines the scope and limitations within which the results should be interpreted.

Chapter 9 concludes the thesis by consolidating the principal conclusions arising from the research, outlining directions for future work, and presenting the final remarks of the thesis.

Chapter 2: Literature review

This chapter is divided into three sections. Section 2.1 reviews the literature on approaches to modelling the airspace environment. The review focuses on data-driven airspace models that use aerospace big data. It particularly addresses the effective management of large-scale datasets, including large-scale air traffic data and data on new entrant air vehicles, while optimizing computational resources. Section 2.2 identifies and describes 18 key observations drawn from the reviewed airspace models that relate to the main research objective of this thesis. Section 2.3 introduces fundamental concepts in machine learning, reviews their use in airspace modelling, and adds 5 additional observations. Section 2.4 synthesizes the 28 detailed observations identified in Chapters 1 and 2 into five summary observations and four research gaps. These gaps provide the basis for the research questions, hypotheses, and validation strategy presented in Chapter 3.

2.1 Background on airspace models

One way to leverage big data in the aerospace field is to model the airspace environment. This section presents the literature on data-driven approaches to modelling the airspace environment.

Table *3* summarizes a selection of model types used, their subtypes, and specific examples. These examples constitute a representative subset of the numerous airspace models reviewed throughout Section 2.1, illustrating the diversity of modelling approaches and applications within the identified model types and subtypes.

Table 3: Airspace models

Model type	Model subtype	Specific models and examples
Congestion prediction models	Micro-level congestion models	• Kalman Filter State Estimation [80] • Multi-Agent Simulation for Traffic Congestion [81]
	Macro-level congestion models	• Monte Carlo Traffic Prediction [81] • Probabilistic State Prediction [82]
Predictive trajectory models	Individual trajectory models (micro-level)	• Probabilistic 4D trajectory prediction [82] • Deep reinforcement learning for prediction [83]
	Aggregate trajectory models (macro-level)	• Hidden Markov model for long-range prediction [84] • ADS-B-based Trajectory Models [80]
Conflict detection & resolution (CD&R) models	Tactical CD&R models (real-time)	• ORCA-based collision avoidance [85] • Multi-aircraft conflict detection for 4D operations [86] • Monte Carlo Simulations for UAV CD&R [87]
	Strategic CD&R models (pre-flight planning)	• Pre-flight UAV traffic management [57] • Spatio-temporal conflict avoidance [81]
Flow control models	Sector-based flow models	• Sector-based traffic CD&R [86] • High-density UAV air traffic flow [87]
	Dynamic flow models	• Dynamic grid partitioning for CD&R [81]
Airspace weather models	High-fidelity weather models	• Weather uncertainty in CD&R [52] • Monte Carlo simulation for weather impact [86]
	Simplified weather models	• Simplified state estimation with weather inputs [80] • Dynamic airspace configuration under weather constraints [88]
Geospatial airspace models	Static geospatial models	• Static grid-based airspace simulation [81] • Airspace geofencing for urban UTM [89]

	Dynamic geospatial models	• Dynamic geofencing for UAS traffic [90] • 3D Monte Carlo for UAS collision risk [91]
System-wide simulation models	Policy simulation models	• Hybrid-system simulation for NAS safety [92] • Performance-based urban airspace model [76]
	Operational simulation models	• Multiscale dynamic planning for drone traffic [93] • Hybrid UAV integration simulations [94]
UAS models	Localized UAS models (micro-level)	• UAV path prediction for manned aircraft [95] • 3D collision-free path planning [96]
	Regional UAS models (macro-level)	• Dynamic geofencing for low-altitude airspace [90] • Monte Carlo risk assessment for UAS integration [91]
Hybrid models	-	• Hybrid-system simulation for NAS safety [92] • Hybrid intelligent UAV navigation [97]

2.1.1 Congestion prediction models

Congestion prediction models are airspace models designed to forecast congestion in air traffic networks by analyzing factors that influence traffic density, airspace capacity, and delay propagation [70]. These models are important for traffic flow management (TFM), conflict detection, and delay mitigation strategies. Figure 4 summarizes the main components of these models, including their inputs, processing techniques, and outputs. Their inputs can include historical air traffic data (recorded flight trajectories, flight schedules, and congestion levels) [98], real-time surveillance data (ADS-B, radar tracking, and controller inputs) [80], weather conditions (turbulence, wind, and visibility levels affecting air traffic movement), and sector and airspace capacity constraints (maximum allowable aircraft per sector). A variety of processing techniques are employed in the literature for developing congestion prediction models. Shone et al. [98], for example, demonstrate the use of mathematical models and optimization techniques, including stochastic models, queuing theory, and flow-based models, to predict and mitigate congestion. Wanke et al. [54], on the other hand, make use of probabilistic models that estimate uncertainty in air traffic demand, congestion severity, and resolution strategies. A third processing technique to estimate congestion is to use machine learning and AI algorithms (decision trees, random forests, and deep learning models), as shown in the research done by Zhao et al. [99]. The outputs of congestion prediction models produce a variety of evaluation metrics for understanding the state of congestion in a given region of the airspace. These include: predicted congestion levels in specific airspace regions (high-density terminal areas or en-route airspace), recommended traffic rerouting or delay management strategies [100], and estimated flight delay impacts based on congestion forecasts [101].

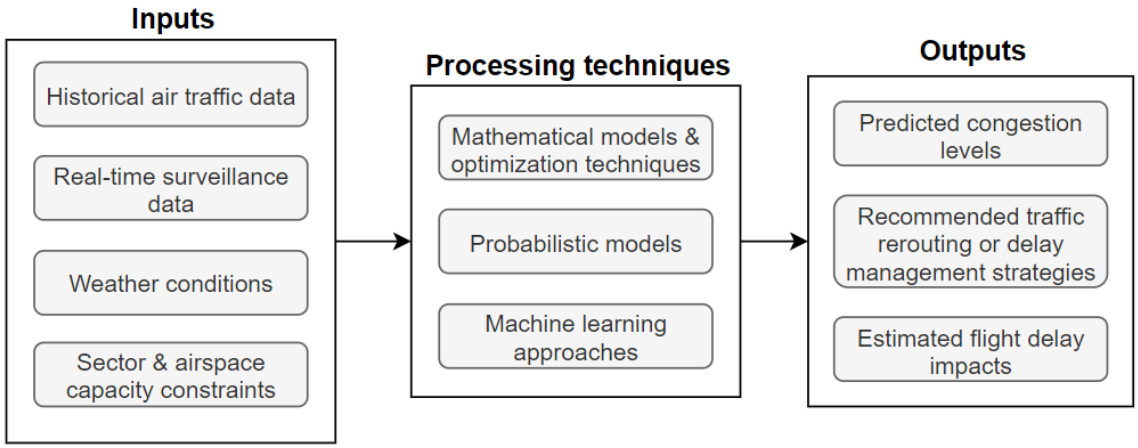


Figure 4: Key components of congestion prediction models

Congestion prediction models have several practical applications in air traffic management (ATM) and aviation decision support systems: Sutton and Lindsay [102] show their use in Traffic Flow Management (TFM) and airspace capacity planning to help predict congestion hotspots and enable proactive air traffic control interventions to balance workload across different airspace sectors. Yang et al. [103] demonstrate how predicting congestion is essential for detecting and mitigating potential airspace conflicts, particularly in high-density terminal areas and en-route airspaces. Congestion prediction models are not only used for air traffic operations but also airport ground operations, as demonstrated by Lee and Balakrishnan [104] in a model used to predict taxiway and runway congestion at busy airports, allowing for more efficient aircraft scheduling and sequencing. Recent research focuses on congestion prediction for new airspace entrants, such as drones and eVTOL aircraft. Alharbi et al. [53] propose a deep learning-based model to forecast UTM air traffic congestion, considering no-fly zones, recreational drone activity, and environmental constraints.

Many congestion prediction models rely on aerospace big data, particularly from surveillance, navigation, and airline operations. However, the extent to which big data is used varies across models:

- Models that use real-time big data: Some models incorporate ADS-B and radar data to perform real-time congestion monitoring and trajectory predictions. Xi et al. [80] demonstrate how ADS-B-based trajectory prediction algorithms improve congestion forecasting in high-density airspace.
- Models using historical big data for training AI algorithms: Machine learning-based models, such as those by Zhao et al. [99], train AI models using historical air traffic data, enabling the prediction of congestion levels in different air traffic regions with up to 86.6% accuracy.
- Models combining historical and simulated data: When real-world data is insufficient (for modelling future UAM and drone congestion), researchers rely on synthetic datasets and simulation frameworks. Shresta et al. [105] use a fast-time NAS simulation model to analyze congestion patterns for projected UAM operations.
- Models that do not use any aerospace big data: Some types of congestion prediction models such as Odoni and Simpson's [106] queuing model use analytical/equation-based approaches and do not use any data to predict congestion. Other examples of this include the integer optimization model developed by Bertsimas and Patterson [107] or Wanke et al.'s [54] rule-based congestion assessment model. This thesis only focuses on airspace models that make use of big aerospace data and thus these models are considered out of the scope.

Marzuoli et al. [108] highlight that large-scale NAS congestion modelling requires very high computational power, requiring optimized algorithms and network synthesis techniques to handle the complexity of air traffic flow data.

2.1.2 Predictive trajectory models

Predictive trajectory models are airspace models designed to forecast the future positions and paths of aircraft based on various influencing factors [51]. These models are fundamental for ATM, conflict detection, and trajectory-based operations (TBO). Figure 5 summarizes the main components of these models including their inputs, processing techniques, and outputs. Inputs can include aircraft state data (position, speed, altitude, heading, and acceleration usually obtained from surveillance technologies like ADS-B and radar [51]), weather conditions (wind speed, turbulence, temperature, and atmospheric pressure affecting aircraft performance [109]), flight plans (airline-filed schedules and route constraints), and historical trajectories [110]. Several processing techniques are employed in the literature for developing predictive trajectory models. Hunter [111] employs a physics-based approach by using kinematic and aerodynamic equations to predict aircraft movement based on thrust, drag, and lift forces. Casado Magaña

and Juan [112] used a probabilistic approach to predict aircraft trajectories by utilizing stochastic methods like Monte Carlo simulations and Polynomial Chaos Expansions (PCE) to account for uncertainty in trajectory predictions. Gallego et al. [113] showed that machine learning approaches could also be used to model aircraft trajectories by applying deep learning models, LSTMs, and random forests to learn from historical data and improve prediction accuracy. The outputs of predictive trajectory models include 4D predicted trajectories (future aircraft positions in latitude, longitude, altitude, and time [82]), trajectory uncertainty estimates (probability distributions of deviations from predicted paths), and conflict predictions (detection of potential air traffic conflicts and separation violations [84]).

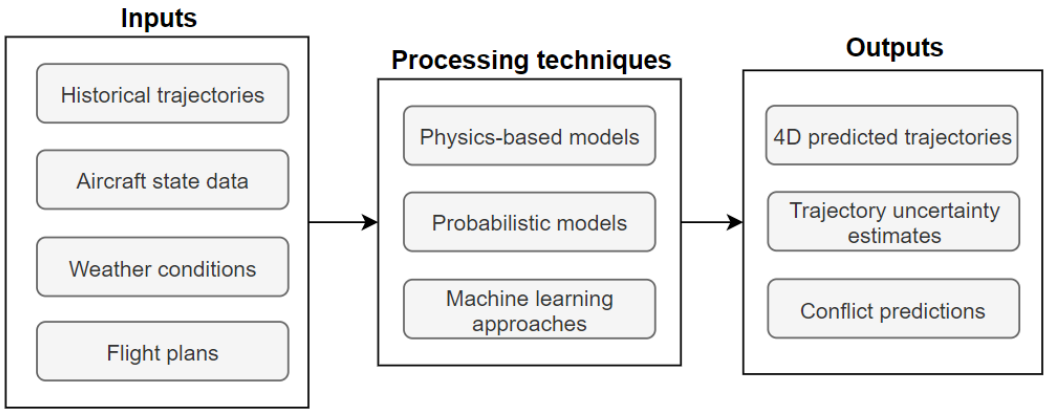


Figure 5: Key components of predictive trajectory models

Predictive trajectory models are used in applications such as Air Traffic Flow Management (ATFM), conflict detection and avoidance, UAS Traffic Management (UTM), weather impact assessments, and autonomous flight operations. Kim et al. [114] optimize airspace capacity utilization for ATFM applications by predicting where congestion is likely to occur and adjusting flight routes accordingly. Liu and Hwang [82] developed a probabilistic 4D trajectory prediction model to help controllers detect and resolve conflicts before they occur to reduce mid-air collision risks. Murca and Oliveira [109] investigated the use of predictive trajectory models to study safe drone and UAM operations in a shared airspace. Kim et al. [115] use predictive models for anticipating weather disruptions and dynamically rerouting flights. Asadi and Richards [116] describe the use of predictive trajectory models for autonomous flight operations to ensure safe and efficient flight paths.

Predictive trajectory models rely heavily on big aerospace data, sourced from real-time and historical air traffic surveillance systems. The extent to which these models use big data varies across approaches:

- Real-time data integration: ADS-B and radar-based trajectory predictors rely on very large datasets to continuously refine trajectory forecasts [115].
- Historical big data for machine learning models: Deep learning-based trajectory predictors require large historical flight datasets to train models and improve their accuracy [110].
- Hybrid approaches combining real-time and historical data: Some models merge real-time sensor data with historical trends to enhance predictions [109].
- Models that do not use any aerospace big data: Physics-based (kinematic and aerodynamic) models [111] and geometric-based models [117] (which use spatial proximity and vector calculations to estimate future positions of aircraft) are predictive trajectory models that do not use aircraft data. This thesis only focuses on airspace models that make use of big aerospace data and thus these models are outside of the scope of this thesis.

Gallego et al. [113] explain that handling large-scale trajectory data requires optimized computing infrastructures. Cloud-based processing and parallel computation techniques have been explored to improve efficiency [113].

Predictive trajectory models can also be classified into two broad categories: individual and aggregate. Individual trajectory models focus on predicting and managing the path of a single aircraft. These models rely on precise aircraft dynamics, real-time sensor data, and probabilistic methods to estimate the most likely future trajectory of a specific flight. Aggregate trajectory models focus on group-level or systemic behaviour, predicting the movement of multiple aircraft in a networked airspace system. These models are primarily used for traffic flow management, congestion forecasting, and large-scale airspace planning. Table 4 summarizes the key differences between individual and aggregate predictive trajectory models.

Table 4: Key differences between individual and aggregate predictive trajectory models

Feature	**Individual trajectory models**	**Aggregate trajectory models**
Focus	Single aircraft trajectory	Group-level/system-wide flow
Granularity	High (detailed aircraft motion)	Lower (statistical/network-based approach)
Time horizon	Short-term (seconds to minutes)	Medium to long-term (hours to days)
Computational cost	High (real-time updates)	Moderate to low (simplified abstractions)
Application	Conflict detection, ATC decision support, drone operations	Traffic flow management, congestion forecasting, airspace design
Example model	Murphy et al.'s [118] physics-based kinematic model	Sawhill et al.'s [119] multi-aircraft network model

2.1.3 Conflict detection and resolution models

Conflict detection and resolution (CD&R) models are designed to identify and mitigate the risk of mid-air collisions or other operational conflicts between aircraft [117]. These models are important to modern ATM, especially as airspaces becomes more congested. CD&R models work by monitoring aircraft trajectories and applying predictive algorithms to detect potential conflicts, followed by the recommendation or automatic execution of maneuvers to resolve the conflict [117]. Figure 6 summarizes the main components of these models, including their inputs, processing techniques, and outputs. Inputs to CD&R models include aircraft state data (position, velocity, heading, altitude) from ADS-B, radar, or surveillance systems [120], flight intent information (filed flight plans, air traffic control (ATC) clearances) for pre-planned trajectories [117], and weather data (turbulence, wind, convective activity) affecting aircraft paths [121]. Four different processing techniques or modelling approaches exist for CD&R models: Geometric, probabilistic, rule-based, and machine learning-based CD&R models. Geometric CD&R models use geometric relationships between aircraft positions and trajectories to detect conflicts [117]. The model calculates when two aircraft will violate minimum separation standards and generates avoidance maneuvers based on geometric principles like heading, altitude, or speed adjustments [117]. Probabilistic CD&R models predict the likelihood of a conflict based on uncertainty in trajectory predictions such as uncertainty due to weather or aircraft performance variability [122]. They do not just rely on deterministic projections but instead assign probabilities to future positions and calculate the probability of a conflict occurring. For example, Monte Carlo Simulations can be used to model multiple possible future trajectories with various uncertainties to estimate the risk of conflicts [123]. Rule-based CD&R models rely on predefined separation rules and regulations such as those established by aviation authorities (TCCA, FAA, EASA) to detect conflicts and propose solutions [124]. These models use transparent decision-making

processes based on compliance with regulatory standards [124]. One example of a rule-based CD&R model is the Traffic Alert and Collision Avoidance System (TCAS), which is a real-world rule-based system that provides vertical resolution advisories to pilots when potential conflicts are detected [125]. Machine learning-based CD&R models are newer models that rely on or make use of machine learning, where algorithms learn from historical flight data and conflict scenarios to improve predictions and optimize resolution strategies [126]. These models improve their accuracy over time by continuously updating their predictions based on new data [126]. Reinforcement Learning has been used in CD&R systems to train models on how to autonomously resolve conflicts in dynamic environments without human intervention [127], [128]. The outputs of CD&R models include conflict alerts (indicating a loss of separation risk) and resolution recommendations (new trajectory, altitude change, rerouting).

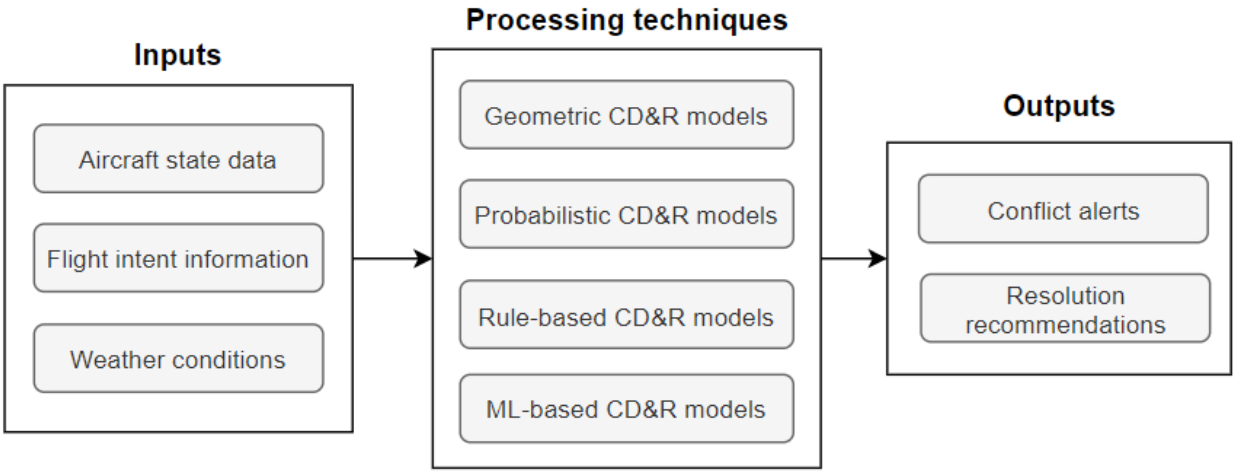


Figure 6: Key components of CD&R models

CD&R models can be classified based on their operational phase (tactical real-time vs. strategic pre-flight planning) or based on the modelling approach (geometric, probabilistic, rule-based, or machine learning-based). Tactical CD&R models, such as the ORCA-based Collision Avoidance method, dynamically resolve conflicts during flight by calculating avoidance maneuvers in real time [85]. In contrast, strategic CD&R models identify conflicts and generate conflict-free paths before departure, often using spatial geofencing techniques to manage reserved airspace [81]. Table 5 summarizes the main differences between the model types.

Table 5: CD&R models classified based on operational phases

CD&R type	Description	Example models
Tactical CD&R (Real-time)	Operates in real-time to prevent imminent conflicts within minutes	Neural network-based real-time detection [126]
Strategic CD&R (Pre-flight planning)	Prevents conflicts in advance by optimizing trajectories before departure	Graph-based airspace optimization [129]

Table 6: CD&R models classified based on modelling approach

Model type	Description	Example references
Geometric CD&R	Uses deterministic spatial calculations (closest point of approach (CPA)) to predict conflicts	Chiang et al. introduced a prioritized geometric routing method [117].
Probabilistic CD&R	Uses uncertainty modelling (Monte Carlo simulations) to predict likely conflicts based on aircraft motion variance	Prandini et al. proposed a probabilistic approach for near-term conflict detection [121].
Rule-Based CD&R	Uses predefined separation standards (TCAS algorithms) to suggest maneuver actions	Geser et al. developed a recovery model for conflict-free rerouting [130].
Machine Learning-Based CD&R	Uses neural networks, reinforcement learning, and AI-based policy models for conflict resolution	Xu et al. introduced a reinforcement learning-based CD&R algorithm [131].

CD&R models increasingly rely on big aerospace data for improving accuracy and predictive capabilities. The extent of big data usage varies by approach:

- Data-driven CD&R models (machine learning-based models) train AI models on historical flight data, surveillance data (ADS-B, radar), and weather conditions. Bastas and Vouros [132] showed that AI models trained on ATC actions can predict human-like conflict resolutions.
- Tactical CD&R models integrate real-time data like ADS-B tracking, sensor fusion, and ATC inputs for immediate conflict detection [126].
- Strategic C&R models use historical trajectory distributions and Monte Carlo simulations for probabilistic modelling [121].

2.1.4 Flow control models

Flow control models are airspace models used in ATM to help regulate aircraft movements to maintain efficient, safe, and conflict-free airspace operations. These models balance air traffic demand with available capacity and prevent congestion while optimizing airspace utilization [133]. Figure 7 summarizes the main inputs, processing techniques, and outputs of flow control models.

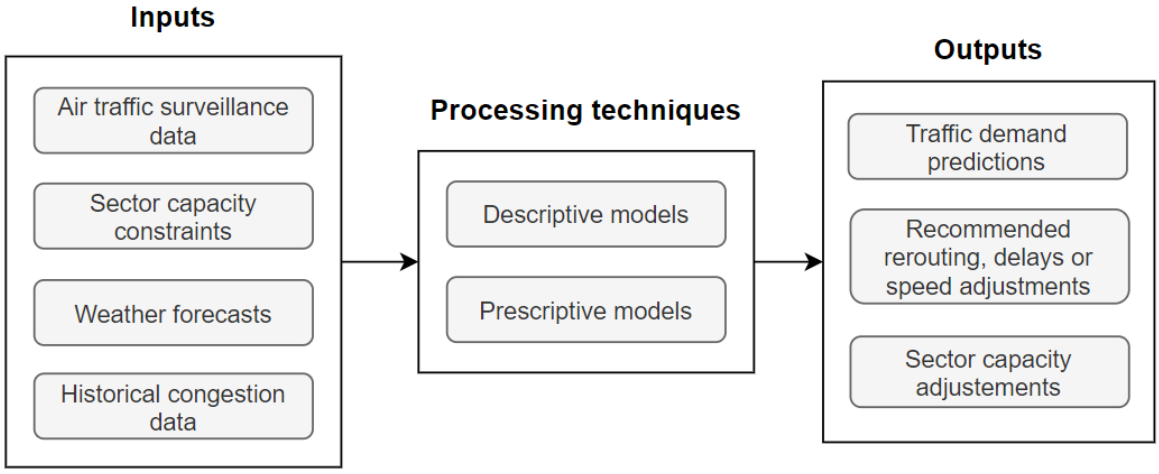


Figure 7: Key components of flow control models

Inputs include air traffic surveillance data (ADS-B, radar, flight plans), sector capacity constraints (maximum aircraft per region, air traffic controller workload), weather forecasts (convective weather, turbulence, visibility), and historical congestion data (patterns of delays and sector overloading).

The literature reviews performed by Sun [134] and Sridhar et al. [135] group different flow control models in the following way:

1. **Descriptive models**: These models are concerned with understanding, predicting, and assessing the state and evolution of air traffic flow in the NAS using physics-based equations [135].
   1.1. **Lagrangian models** (Aircraft-level models / trajectory-based models): These models predict the behaviour of the NAS by propagating the trajectories of individual aircraft forward in time. They are used to count the number of aircraft in a region of airspace. Examples of systems using this approach include the Center TRACON Automation System, the Future ATM Concepts Evaluation Tool (FACET), and the Collaborative Routing Coordination Tool [135], [136].
   1.2. **Eulerian models** (Aggregate models / control-volume based): Designed for strategic Traffic Flow Management (TFM), these models control flows of aircraft rather than individual aircraft by spatially aggregating traffic flow, which makes them computationally efficient when compared to aircraft-level models like Lagrangian models [134]. They simplify the analysis and design of complex systems [134]. General Eulerian models (e.g., 2D spatial aggregation) [137], the Linear Dynamic System Model (LDSM) [138], Partial Differential Equation (PDE) models [139], and Cell Transmission Model (CTM)-inspired models [140] are all different subtypes of Eulerian models using different aggregation approaches.
2. **Prescriptive models**: These models focus on formulating the TFM problem as a mathematical optimization problem for determining control actions that achieve desired system performance, such as minimizing delay or maximizing throughput, subject to constraints [135].
   2.1. **Linear Programming (LP) and Integer Programming (IP) / Mixed Integer Linear Programming (MILP)**: The TFM problem is formulated as a binary Integer Program (IP) [141], a mathematical optimization problem where some or all decision variables must take integer values, or Mixed Integer Linear Program (MILP) [142], where some variables are required to be integers (often binary) and the objective function and all constraints are linear. These models determine optimal decision variables such as departure (ground) delay, rerouting, and airborne holding for individual flights or groups of flights [134].
   2.2. **Dynamic Programming (DP)** [135]: DP offers an approach to solve the TFM optimization problem, which can be formulated as a Two-Point Boundary Value Problem (TPBVP), by optimizing over multiple stages or time intervals.
   2.3. **Adjoint-based Optimization** [143], [144]: This method is used in conjunction with descriptive models like the PDE model to efficiently evaluate the gradient of the objective function. This gradient information is then used by nonlinear optimization methods to find optimal control strategies, such as adjusting allocation matrices for flow redirection
3. **Machine learning models**: Aditya et al. [145] reviewed the state of flow control models in 2024 and they observed a shift toward real-time dynamic optimization as the future direction for flow control models, in support of newer, data-driven modelling paradigms (like machine learning) that complement but differ from traditional mathematical formulations. Sridhar et al. [146] explicitly contrasted physics-based (traditional) methods with machine learning techniques, emphasizing their complementary roles. They provided case studies on conflict prediction, delay/cancellation forecasting, and reroute advisories—demonstrating how ML augments both descriptive and prescriptive modelling tasks. Their study recognized that ML-based solutions sit alongside classic optimization and simulation approaches

Outputs of flow control models include traffic demand predictions (expected congestion in different airspace regions), recommended rerouting, delay assignments, or speed adjustments to balance traffic, and sector capacity adjustments (either keeping static sector boundaries or dynamically altering them).

While most flow control models incorporate large-scale aerospace data to enhance their predictive accuracy, the extent of their reliance on such data varies across modelling approaches:

- Models highly dependent on big data include real-time AI-based flow models which process live ADS-B surveillance data, controller workload information, and congestion forecasts. Menon et al. [137], for example, developed an Eulerian-based air traffic flow model, integrating real-time traffic observations with predictive control strategies.
- Models using moderate amounts of historical big data for forecasting include machine learning-based traffic demand prediction models trained on past congestion data to anticipate high-demand areas. Alharbi et al. [147] trained an AI-driven airspace density estimation model on historical drone movement patterns to predict high-traffic urban air mobility zones
- Models that require minimal big data include mathematical optimization models (Eulerian and Lagrangian) that operate on theoretical air traffic flow constraints rather than large datasets. Bayen et al. [148] developed a flow-based optimization model using partial differential equations (PDEs) to simulate air traffic dynamics without real-world data

2.1.5 System-wide airspace simulation models

System-wide airspace simulation models are designed to simulate the entire National Airspace System (NAS), including not only aircraft operations but also components such as weather, navigation aids, air traffic controllers, and pilots [149]. These models are valuable for evaluating the impact on the overall airspace system of operational changes or disruptions such as weather events or increased traffic volume.

Figure 8 summarizes the main inputs, processing techniques, and outputs of system-wide airspace simulation models. Inputs include air traffic surveillance data (ADS-B, radar, flight schedules, flight trajectories), weather conditions (convective weather patterns, wind conditions, turbulence forecasts), navigation and airspace structure data (sector definitions, separation standards, routes), and air traffic control (ATC) interventions (conflict resolution, rerouting, speed adjustments). Three different modelling approaches can be used to develop system-wide airspace simulation models: Agent-Based Modelling (ABM), stochastic models, or hybrid systems modelling. ABM simulates individual aircraft, air traffic controllers, or automation systems as independent entities, or "agents," each following a set of behavioural rules. These agents interact within the airspace environment, enabling the study of emergent traffic patterns, controller workload, and congestion effects. Wei and Li [150] introduced a multi-agent simulation framework where aircraft agents and ATC agents cooperated to manage traffic flow, thereby demonstrating improved realism in air traffic simulations. Wang et al. [151] analyzed air traffic flow control using an agent-based approach, showing how individual aircraft behaviour leads to congestion and delay patterns. Stochastic models incorporate randomness to account for uncertainty in air traffic operations. These models simulate weather effects, flight delays, congestion, and ATC workload variations by assigning probabilities to different system states. Glover and Lygeros [152] introduced a stochastic hybrid air traffic control model, integrating random wind fields and flight uncertainty into conflict detection and resolution simulations. Seah and Hwang [153] proposed a stochastic linear hybrid system for aircraft tracking and traffic management, improving prediction accuracy in high-density airspace. Hybrid system models combine deterministic and stochastic approaches to simulate real-time air traffic dynamics. They integrate Discrete-event simulation (DES) for controller actions and ATC interventions with continuous-time models that simulate aircraft movement using physics-based equations. They can also include probabilistic components

to account for uncertainty in weather conditions, flight delays, and sector congestion. Kamgarpour [154] developed an optimal control framework for air traffic flow, integrating stochastic weather modelling with deterministic flight trajectory optimization. Medianto et al. [155] introduced a hybrid air traffic simulation model that combines agent-based simulation, discrete event modelling, and system dynamics to allow for a realistic simulation of Terminal Control Area operations. The outputs of system-wide airspace simulation models include predicted congestion levels and airspace sector utilization rates, optimal routing and scheduling strategies for reducing flight delays, and impacts of inclement weather and disruptions on airspace performance.

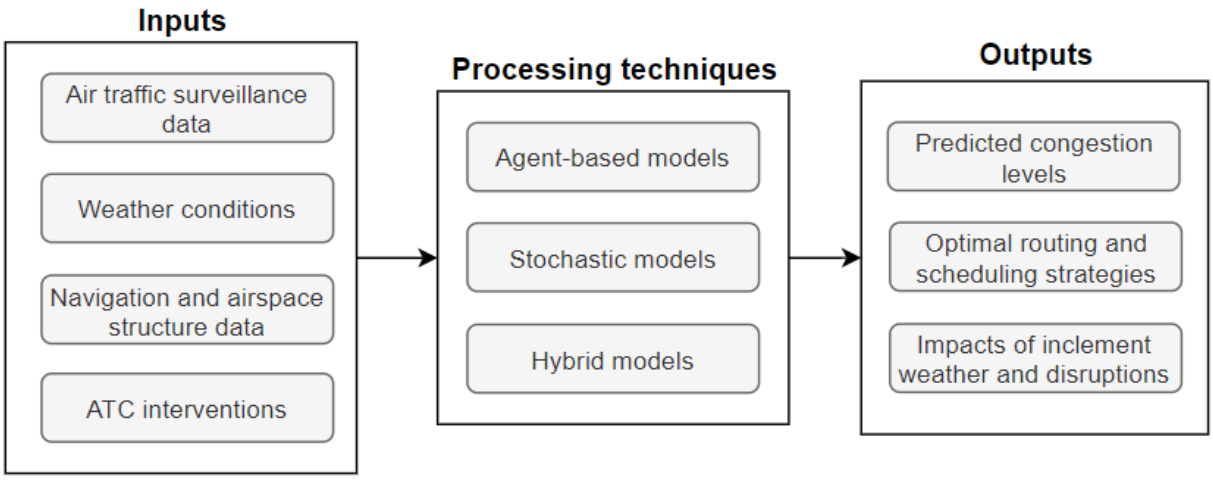


Figure 8: Key components of system-wide airspace simulation models

Table 7: Summary of the different system-wide airspace simulation models

Processing technique	How it works	Example use case	Big data dependency
Agent-Based Modelling (ABM)	Models individual aircraft and controllers as independent entities.	Wei and Li [150]: ATC multi-agent simulation.	Moderate: Relies on behaviour rules, not large-scale data.
Stochastic models	Simulates random events (weather, delays, congestion).	Glover and Lygeros [152]: Stochastic wind field modelling.	High: Needs real-world statistics, historical delays.
Hybrid system modelling	Combines discrete-event, physics-based, and probabilistic approaches.	Medianto et al. [155]: Hybrid simulation for Terminal Control Area.	High: Uses AI models, ADS-B tracking, real-time data.

System-wide airspace simulation models are mainly used for three practical applications: air traffic flow and congestion management, studying the impact of inclement weather on airspace operations, and next-generation air traffic management and airspace design. For air traffic flow and congestion management, McGovern and Kalish [156] developed a Monte Carlo-based stochastic simulation tool to model airspace capacity, ATC workload, and flight rerouting strategies. Krozel & Doble [157] simulated the U.S. NAS under inclement weather conditions to study air traffic flow constraints and Roy et al. [158] proposed a stochastic network model to simulate the propagation of weather-induced delays across the airspace system. In the context of next-generation air traffic management and airspace design, Pasaoglu et al. [159] introduced a hybrid system model for automated air traffic control (ATC) separation assurance, validating

it on real air traffic data. Baspınar et al. [160] developed an autonomous air traffic control system to handle increasing air traffic demand using integer linear programming-based trajectory prediction.

The extent to which system-wide airspace simulation models rely on big aerospace data varies depending on the modelling approach.

- Models highly dependent on big data: Real-time and AI-driven models use large-scale datasets, including ADS-B data for live tracking of aircraft movements and ATC logs and weather reports for congestion and rerouting simulations. For example Sridhar et al. [161] developed the Future Air Traffic Management Concepts Evaluation Tool (FACET), which simulates nationwide air traffic using real-time flight tracking data.
- Models using historical big data for forecasting: Machine learning-based air traffic models rely on past airspace congestion patterns to predict future traffic trends. For example, Tian et al. [162] introduced a stochastic simulation optimization model to evaluate different flight rerouting strategies based on historical air traffic delay data.
- Models that require minimal big data: Mathematical optimization models use predefined airspace constraints and do not rely heavily on empirical datasets. For example, Kamgarpour [154] developed an optimal control model for air traffic flow by integrating stochastic weather impacts without using large-scale real-time data.

2.1.6 Geospatial airspace models

Another approach to modelling the airspace makes use of Geographic Information Systems (GIS) information. Geospatial airspace models represent airspace entities, structures, and constraints within a common geographically referenced framework. The United States Geological Survey (USGS) defines a GIS as an "organized collection of computer hardware, software, and geographic data designed to efficiently capture, store, update, manipulate, analyze, and display all forms of geographically referenced information" [163]. For clarity, this thesis will refer to airspace models that organize and analyze geographically referenced airspace data as *geospatial models*. Geospatial models represent the airspace and its constituent parts as a *geospatial system*, where each element present in the model has a 3D location that can change over time, and is associated with a GIS reference system [164], [165], [166], [167].

Figure 9 shows a summary of the inputs, processing techniques, and outputs of geospatial airspace models. Inputs to geospatial airspace models include geospatial data layers (digital elevation models, terrain data, airspace boundaries, etc.), flight trajectory data (historical and real-time aircraft movement data from ADS-B or radar), aeronautical information (navigation aids, obstacle databases, weather patterns), urban infrastructure datasets (building heights, restricted airspace, UAM corridors). Geospatial airspace models can make use of three different types of data processing techniques to analyze, visualize, and manage airspace operations: GIS-based spatial modelling, Discrete global grid systems (DGGS), and 3D and 4D visualization techniques.

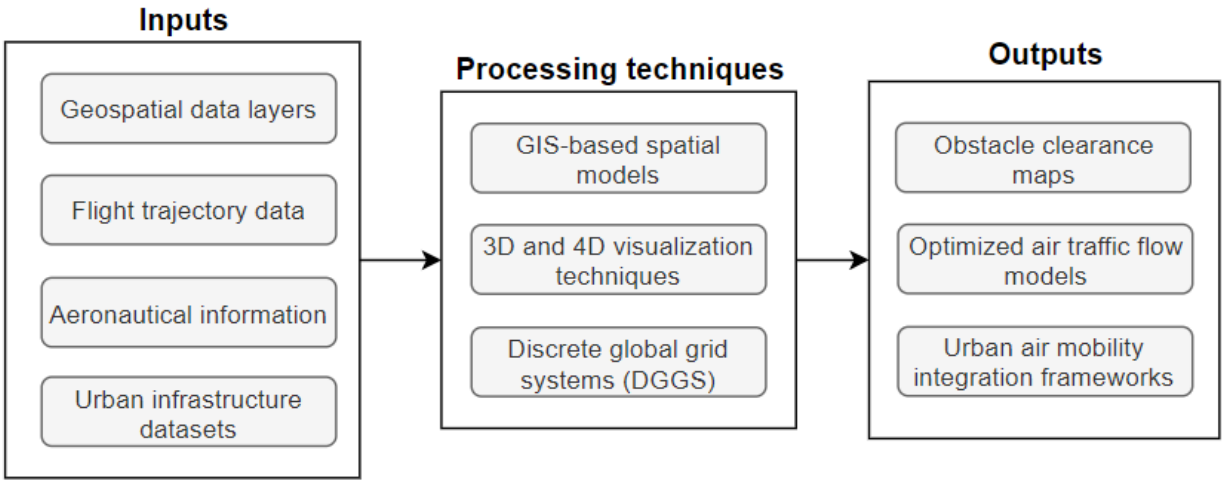


Figure 9: Key components of geospatial airspace models

GIS-based spatial modelling integrates geospatial data, airspace structures, and traffic flow data to create maps, models, and decision support systems. This technique allows overlaying multiple layers of geospatial information, such as terrain elevation, restricted airspace, and air traffic routes, to support airspace management. For example, Mitsevich and Zhukovskaya [165] demonstrated how GIS-based spatial analysis is used for aerodrome land development and airspace zoning, ensuring compliance with aviation safety standards. Chang [167] proposed a GIS obstruction model that analyzes airspace safety near airports, ensuring that buildings and structures comply with regulatory height restrictions. Mekni [168] introduced a GIS-based virtual geospatial environment (VGE) that integrates topological relationships between airspace layers, allowing air traffic controllers to manage flights more efficiently.

A second processing technique used to develop geospatial airspace models uses Discrete Global Grid Systems (DGGS) [169], [170], which is a spatial indexing system that divides the Earth's surface into a hierarchical set of cells, allowing scalable, multi-resolution airspace modelling. In contrast with conventional planar grids, DGGS frameworks support consistent spatial indexing over large geographic extents and can reduce problems associated with projection discontinuities and distortions [171]. Li and Stefanakis [172] compared DGGS operations with traditional GIS, highlighting DGGS advantages in multi-resolution airspace analysis and 3D spatial indexing. Djavaherpour et al. [173] demonstrated how DGGS can be used for physical visualization of global geospatial datasets, improving airspace data management. Jie-qing [174] introduced Sphere Degenerated Octree Grid (SDOG), a DGGS variant that provides a hierarchical, multi-resolution 3D spatial representation of airspace data without projection distortion. Within the aerospace field, DGGS and related hierarchical global grid structures have been applied to several specific functions. Deng et al. [175] developed the GeoSOT-AS framework to organize, index, and retrieve ADS-B observations within a four-dimensional airspace–time grid. Zhai et al. [176] used GeoSOT-3D cells to transform UAV collision detection from pairwise distance calculations into multilevel spatial queries. Sahadevan Neelakandan and Al Ali [177] represented UAV trajectories through hexagonal grid indexing to support the integration of trajectory-based UTM and ATM operations. Zhai et al. [178] also used global subdivision grids to construct a low-altitude public air-route network for UAV management, while Liu et al. [179] incorporated multi-resolution grids into an airspace-sectorization framework. These studies demonstrate that hierarchical grid structures can support aviation-data organization, trajectory representation, spatial searching, conflict detection, route-network construction, and airspace sectorization.

A third processing technique used to develop geospatial airspace models employs 3D and 4D visualization techniques which extend traditional GIS by adding three-dimensional airspace representations and temporal dynamics. These models allow for aircraft trajectory visualization, altitude-aware air traffic control, and real-time airspace monitoring. Francois et al. [180] developed ISO-standard 3D air traffic models that allow air traffic controllers to analyze aircraft positions, separation minima, and flight levels in three-dimensional space. Panchal et al. [181] created a GIS-based 3D modelling platform, which enables airspace planners to visualize navigation routes, airport layouts, and terrain features. Li et al. [182] introduced 4D-SAS, a dynamic 4D spatial agent-based simulation system that integrates real-time geospatial data for air traffic control and congestion analysis. Chailungka et al. [183] similarly introduced the Geo-Informatic System for Aviation, which incorporates 3D and 4D geospatial analysis into civil aviation planning.

Together, these approaches enable geospatial airspace models to generate outputs such as obstacle-clearance and safety-assessment maps, spatial representations of airspace structures and traffic flows, dynamic aircraft-trajectory visualizations, congestion and route-analysis products, and planning frameworks for conventional and emerging airspace operations. Their principal value lies in integrating heterogeneous spatial datasets within a common representation and supporting applications in air traffic management, airport and airspace planning, aviation safety, and low-altitude operations.

Geospatial airspace models vary in their reliance on big aerospace data, with some using real-time tracking data and others relying on predefined spatial datasets. The extent of this reliance depends on whether the intended application is primarily real-time, historical, or based on static spatial information.

- Models highly dependent on big data: Real-time geospatial models integrate ADS-B data, ATC surveillance feeds, and weather observations. Wright et al. [184] developed AvenGIS, a GIS-based tool for real-time air traffic visualization and ADS-B compliance monitoring.
- Models using historical big data for forecasting: Geospatial models for airspace demand prediction use historical flight density data and congestion trends. Ayhan et al. [70] proposed a big-data-driven airspace congestion prediction model, leveraging NoSQL databases to process large-scale traffic data.
- Models that require minimal big data: Mathematical airspace modelling approaches define airspace constraints and spatial relationships without real-time input. Sun et al. [185] introduced GeoSOT-based airspace grid modelling, which provides a structured approach to representing airspace zones efficiently.

The reviewed studies demonstrate that DGGS and related hierarchical grid structures can support multi-resolution spatial representation, geospatial indexing, visualization, spatial searching, and selected aviation applications. The extent to which these approaches have been adopted in airspace modelling, integrated with heterogeneous aerospace datasets, and reused across different analytical applications is examined in Section 2.2.

2.1.7 Unconventional aircraft airspace models

The rapid development of unconventional aircraft, including UAS, Urban Air Mobility (UAM), Advanced Air Mobility (AAM), eVTOLs, hybrid electric aircraft, and precision agriculture drones, has led to the creation of specialized airspace models to support their safe and efficient integration into national airspace systems. These models [186], [187] differ from traditional airspace models in their ability to handle low-altitude operations, dynamic routing, and mixed-traffic environments. They play an important role in ensuring the safe coexistence of new entrants and existing traditional aircraft, particularly as the use of new entrants continues to expand [188].

Figure 10 summarizes the main inputs, processing techniques, and outputs of unconventional aircraft airspace models.

Inputs to unconventional aircraft airspace models include aircraft performance data (flight dynamics, battery levels for eVTOLs, payload, endurance) [189], airspace structure and rules (navigation corridors, vertiport access, urban low-altitude regulations) [190], traffic flow and demand models (UAS traffic density, arrival sequencing for eVTOLs, congestion) [191], and weather and environmental conditions (wind profiles, urban turbulence, noise constraints) [192].

Three main approaches are used to develop unconventional aircraft airspace models (see Figure 10): simulation-based, optimization-based, and machine learning-based models.

Simulation-based modelling using stochastic simulations are designed to model uncertain and dynamic airspace conditions, incorporating randomized weather effects, flight delays, congestion, and aircraft interactions. This technique is widely used in collision avoidance algorithms, ensuring safe separation between vehicles in high-density airspace. Monte Carlo simulations are commonly employed to estimate collision probabilities and optimize traffic flow in stochastic conditions. McGovern and Kalish [156] developed a stochastic airspace simulation tool for analyzing aircraft operations in the NAS. Their Monte Carlo-based system incorporated weather conditions, air traffic controller decisions, and aircraft behaviour to provide realistic predictions. Gomaa et al. [193] introduced a stochastic model for UAM and UAS collision avoidance, which accounted for wind and wake turbulence effects and was intended to improve separation assurance in urban airspace. Clarke et al. [194] developed a methodology to adjust air traffic based on predicted capacity constraints derived from probabilistic weather forecasts using stochastic airspace capacity models.

Optimization-based models dynamically adjust aircraft trajectories and scheduling to maximize airspace efficiency and safety. They are used in path planning algorithms to minimize fuel consumption, flight time, and congestion. They are also applied in vertiport scheduling to determine optimal landing sequences for eVTOL aircraft in urban airspace. Xie et al. [195] used metaheuristic algorithms such as Tabu search to develop a hybrid AI-based optimization model for dynamic re-routing of UAM and UAS traffic. Their method significantly improved low-altitude air traffic management by resolving conflicts in real-time. Choi and Lee [196] introduced a convex optimization framework for collision-free trajectory planning for AAM aircraft, focusing on avoiding non-cooperative intruders like rogue drones. Mukherjee [197] proposed stochastic dynamic optimization models for air traffic flow management, incorporating real-time capacity constraints and weather conditions.

Machine learning-based models analyze past and real-time flight data to predict airspace congestion, traffic demand, and potential conflicts for unconventional aircraft. Deep learning frameworks can process large-scale datasets from ADS-B, UAS surveillance feeds, and weather inputs to optimize traffic flow; and reinforcement learning (RL) algorithms are used for adaptive air traffic management by enabling real-time adjustments based on evolving airspace conditions. Han et al. [198] developed a deep reinforcement learning (DRL) algorithm for joint velocity and spectrum optimization in UAM networks, demonstrating reduced congestion and improved mission efficiency. Li and Kharchenko [199] applied machine learning to air traffic flow management, optimizing UAS and cargo UAV flight planning under stochastic uncertainties. Murça et al. [200] incorporated AI-driven airspace geofencing and strategic deconfliction to design a stochastic simulation framework for urban AAM air traffic.

Outputs of unconventional aircraft airspace models include separation assurance solutions (ensuring unconventional aircraft maintain safe distances) [196], airspace utilization analytics (identifying congestion

hotspots for drone or UAM traffic) [186], and coordination between conventional and unconventional aircraft (ATM and UTM coordinating) [187].

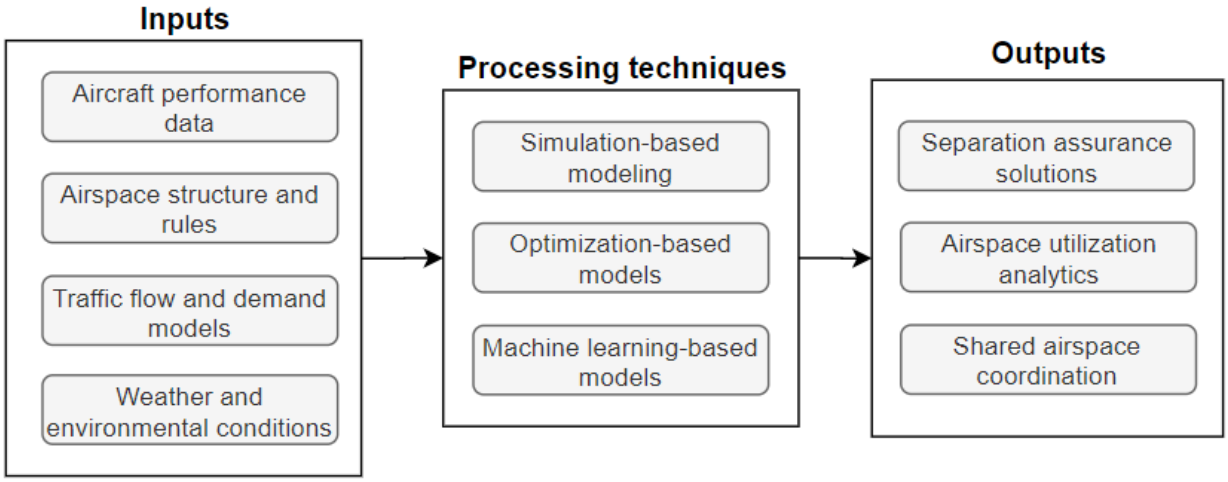


Figure 10: Key components of unconventional aircraft airspace models

Airspace models for unconventional aircraft support various applications in air traffic management, urban mobility, and safety evaluations. These models are used for designing urban air corridors, vertiport operations, and low-altitude traffic management. Song [190] proposed a vertiport airspace control model to optimize landing order for eVTOLs, improving efficiency in high-density urban environments. Bridgelall et al. [201] introduced a deployment framework for eVTOL integration, defining airspace infrastructure requirements, navigation systems, and traffic management solutions. Airspace models for unconventional aircraft also address low-altitude airspace traffic, conflict resolution, and airspace capacity planning for UAS operations. Kopardekar [202] developed a UAS Traffic Management (UTM) framework that enables low-altitude UAS operations, dynamic geofencing, and autonomous separation assurance. Luxhøj [203] introduced a Bayesian risk assessment model for UAS safety in precision agriculture, modelling collision probabilities with manned aircraft. Airspace models for unconventional aircraft also help prevent mid-air collisions and ensure optimal airspace utilization in high-density environments. Choi and Lee [196] proposed an optimization framework for eVTOL collision avoidance, accounting for non-cooperative intruders such as rogue drones. Kochenderfer et al. [204] developed airspace encounter models for unconventional aircraft (gliders, balloons, ultralights) using Bayesian networks to generate statistically valid random trajectories for collision risk assessment.

Airspace models designed for UAS, UAM, AAM, and eVTOL operations typically require large-scale data. However, due to the novelty of these aircraft, historical real-world data is often scarce, leading to the extensive use of simulated data, a distinguishing characteristic of these models when compared to other airspace models, where historical data availability is less of an issue. Shresta et al. [105] discuss how UAS flight plans were developed using simulated flight trajectories due to limited available historical records, demonstrating the reliance on simulated data for modelling UAS integration into the NAS. Borener et al. [205] developed a UAS traffic model for metroplex environments, incorporating stochastic simulations and radar-based simulated air traffic to assess risks associated with unmanned aircraft in urban airspace. The use of simulated data in unconventional aircraft airspace models can be summarized as follows:

- Simulated data for collision avoidance and traffic flow predictions: Simulated data is frequently used in models assessing UAS and UAM safety, particularly for conflict detection, separation

assurance, and detect-and-avoid systems. Dixit et al. [206] integrated real-time flight test data with a simulated air traffic environment to validate UAV collision avoidance systems, demonstrating the use of both real and simulated flight trajectories. Zhao et al. [207] developed a spatiotemporal deep learning model for UAS density prediction, trained entirely on synthetically generated mission scenarios, underscoring reliance on simulated traffic data for density forecasting.

- Simulated data supports airspace traffic forecasting and risk assessments: Because unconventional aircraft operations are highly dynamic and experimental, real-time operational data is often unavailable. As a result, airspace simulation models generate synthetic datasets to evaluate airspace capacity, congestion levels, and risk. Alharbi et al. [147] proposed a UTM traffic flow model based on simulated historical data, identifying airspace congestion patterns from synthetically generated UAS missions. Weinert and Underhill [208] developed a methodology for generating representative small UAS trajectories from open-source geographic data, using synthetic flight paths to produce realistic Monte Carlo collision risk simulations.
- Models that incorporate historical data when available: Despite reliance on simulated data, some airspace models integrate historical data when available, particularly for monitoring UAS operations in controlled airspace. Shresta et al. [209] used historical UAS radar track data to develop a hybrid approach that combines real and simulated flight patterns for airspace modelling. Pothana et al. [210] conducted UAS risk assessments around airports using historical ADS-B aircraft data, though simulated UAS flight paths were still required for probability modelling.

2.1.8 Airspace weather models

Airspace modelling is also used to study airspace weather phenomena. *Airspace weather models* incorporate weather data to predict the impact of adverse weather on airspace capacity and aircraft routing to optimize flight operations.

Figure 11 summarizes the main inputs, processing techniques, and outputs of airspace weather models. Common inputs used in these models include weather data (radar data, convective weather coverage, storm intensity, and cloud tops [51], [211], [212], [213]); air traffic data, including aircraft trajectories; sector boundary crossings; traffic volume [70]; and satellite observations used for the early detection of convective weather and cloud formation [214].

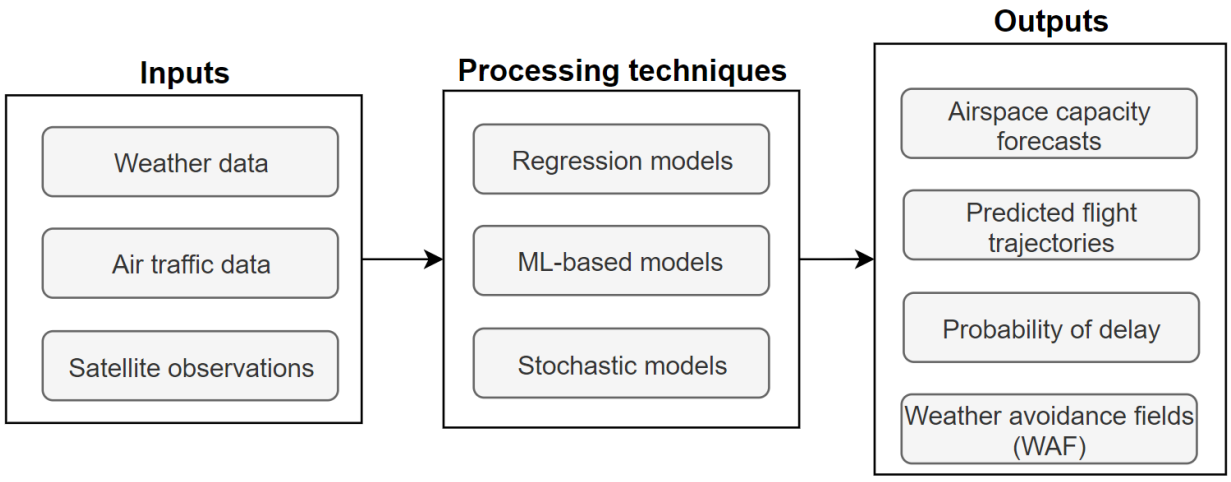


Figure 11: Key components of airspace weather models

Processing techniques employed in the literature for developing airspace weather models include regression models, machine learning models, and stochastic models. Examples from the literature are provided below.

Post et al. [215] developed a multiple regression model that predicted daily airspace system delays. The model utilized input variables such as the number of scheduled arrivals, airport acceptance rates, and an en route weather index that combined lightning strike data with flight plans. This model explained significant day-to-day variability in airspace delays and was able to construct forecasts for air traffic delay patterns. Sridhar and Chen [216] developed a linear autoregressive model that integrated weather-impacted traffic indices and air traffic volume as input variables to predict short-term airspace delays. The model was tailored to predict delays up to two hours ahead and provided adaptive delay estimates, distinguishing between weather-related and non-weather-related causes.

Ayhan et al. [70] developed a data-driven prediction system that employed ML models (linear, non-linear, and ensemble) to predict aircraft counts in airspace sectors. The model took historical aircraft trajectory data, weather conditions, and sector boundary crossings as inputs, efficiently predicting airspace congestion patterns. Ayhan and Samet [51] applied a Hidden Markov Model (HMM) for trajectory prediction. Their model represented airspace as a 3D grid of "weather cubes," where each cube incorporated spatiotemporal data and weather parameters. The HMM framework predicted optimal flight paths by considering environmental uncertainties.

Mitchell et al. [217] designed a stochastic weather model that predicted the maximum airspace capacity by analyzing sector geometry combined with weather constraints. This model calculated the probability distribution of airspace throughput capacity under various weather conditions, thereby improving the reliability of congestion predictions in high-traffic regions. McCrea et al. [218] developed a probabilistic model that integrated stochastic demand and stochastic capacity to predict en route airspace congestion. Their model used severe-weather probabilities from the National Weather Service to determine alternate flight paths and assess potential delays based on probabilistic outcomes.

Outputs of airspace weather models include airspace capacity forecasts (the predicted number of aircraft that can safely pass through an area during adverse weather conditions [217]); aircraft trajectory predictions accounting for weather avoidance strategies [51]; probabilities for predicted delays in air traffic flow due to weather disruptions [216]; and weather avoidance fields (WAF), which are regions in the airspace where pilots are likely to deviate to avoid severe weather [219].

2.1.9 Hybrid airspace models

Hybrid models that combine two or more types of airspace models include congestion prediction, predictive trajectory, conflict detection and resolution, flow control, system-wide airspace simulation, geospatial airspace, unconventional aircraft airspace, and weather models.

Kamgarpour [154] developed a hybrid dynamic model that integrates predictive trajectory control with conflict detection and weather impact assessments. The model used stochastic modelling to capture uncertainty in hazardous weather forecasts. Using a reach-avoid game framework, it predicted safe aircraft paths through unpredictable weather zones, improving conflict detection and flight safety. Liu and Hwang [82] designed a hybrid stochastic linear system (SLHS) that combines 4D trajectory prediction with conflict detection. The model integrated aircraft intent data, weather conditions, and probabilistic path deviation detection to predict conflicts. By approximating conflict probabilities with Gaussian distributions, the model managed airspace congestion and improved prediction accuracy. Huang Ming-hua [50] proposed a hybrid model that combines airspace congestion prediction with flow control models. The model predicted congestion windows and adjusted air traffic flows to minimize delays while balancing equity among

airspace users. Zhou et al. [220] developed a hybrid 4D trajectory prediction model that integrates multiple machine learning techniques. The model combines Kalman filters, LSTM networks, and convolutional neural networks to enhance trajectory predictions under varying weather conditions. By combining these methods across prediction timescales, the model improved prediction accuracy.

2.2 Airspace model literature review

While Section 2.1 presented background information on the fundamental concepts used by the different airspace models found in the literature, Section 2.2 provides a summary of observations from the literature related to the main research objective of this thesis.

2.2.1 Fundamental spatial modelling paradigms

The literature on data-driven airspace models can be classified according to three fundamental spatial modelling paradigms: trajectory-based, region-based, and location-based models, as described below and summarized in Table 8. Figure 12 depicts visual representations of the three approaches.

Trajectory-based models focus on simulating and analyzing the continuous paths of individual aircraft. They emphasize detailed dynamics, interactions, and time-tagged positions. These models are highly dynamic and are typically used in applications requiring precise, real-time assessments, such as conflict detection and resolution. A prominent example is the Traffic Alert and Collision Avoidance System (TCAS) developed by Tang et al. [221], which continuously evaluates distances and directions between aircraft to issue alerts and resolution advisories. Alonso-Ayuso [222] has focused on simulation and optimization strategies for mid-air conflict avoidance by modelling aircraft configurations and trajectories to minimize deviations from flight plans while ensuring safety. Mollinga and Hoof [223] show that the computational requirements of trajectory-based airspace models limit their scalability, making them less suited for analyzing large-scale or long-term airspace scenarios.

Region-based models divide the airspace into spatially bounded zones or sectors, analyzing aggregate traffic behaviour rather than individual aircraft. These models prioritize managing the structure and behaviour of the airspace as a whole, making them well-suited for applications like traffic density management, route planning, or assessing airspace complexity [224]. For example, probabilistic encounter models like the one developed by Artacho [225] are used to calculate risk “hotspots” in the airspace—zones where the probability of collision is elevated—without relying on individual aircraft data. Jang et al. [226] showed that these models are more computationally efficient and scalable when compared to trajectory-based models.

Location-based models, which may use point- or cell-referenced representations, describe airspace conditions at particular spatial locations rather than associating all analytical outputs exclusively with moving aircraft trajectories or aggregating them across broad predefined regions. Dynamic airspace entities, such as aircraft, weather systems, terrain features, and population distributions, may first be represented as spatially and temporally referenced observations. These observations can then be associated with fixed three-dimensional locations or cells and evaluated over time. Examples of location-based airspace models include Zheng et al. [227] developed a spatiotemporal grid model for fine-grained route and resource management. Oh and Yoon [228] associated changing population conditions with spatial locations to support unmanned-aircraft risk restrictions, while Pilko et al. [229] represented dynamic meteorological conditions using time-varying spatial risk maps. These studies demonstrate that location-based formulations can support localized analysis of heterogeneous airspace phenomena.

The distinction between the three paradigms concerns the spatial framework to which analytical outputs are assigned. In trajectory-based models, outputs are associated primarily with individual aircraft paths or interactions between paths. In region-based models, outputs are aggregated over predefined sectors, corridors, or zones. In location-based models, outputs are assigned to fixed spatial locations and time steps.

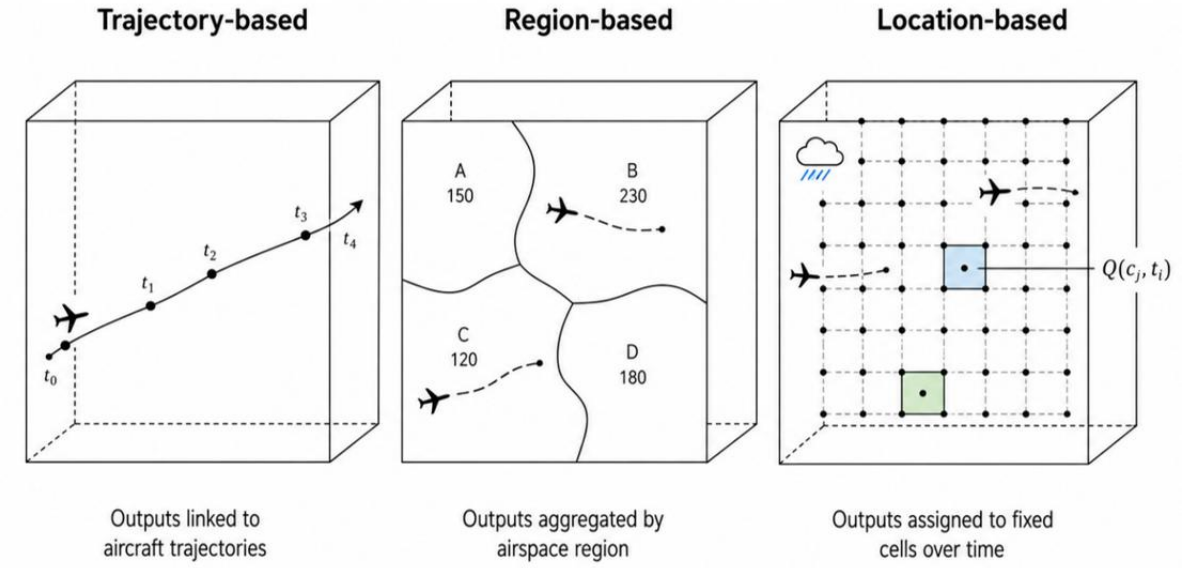


Figure 12: Depictions of the three spatial modelling paradigms

Table 8: Feature comparison of the three spatial modelling paradigms

Feature	Trajectory-based model	Region-based model	Location-based model (point- or cell-referenced)
Modelling focus	Individual aircraft interactions	Aggregated traffic in zones (density)	State or derived property of the airspace at fixed locations
Entity representation	Trajectories (full continuous flight paths)	Zone-level traffic metrics	Dynamic entities associated with fixed 3D locations or cells over time
Data granularity	High (continuous, path-specific)	Medium (zone-aggregated)	Variable (determined by spatial cell size and temporal interval)
Assumptions about vehicles	Requires vehicle type-specific performance characteristics and parameters	Uses class-based abstractions	Core spatial representation can be vehicle-agnostic; application-specific analyses may still require aircraft attributes or performance information.
Data integration capability	Low (requires harmonized formats and intent prediction)	Medium (overlay of spatial data layers)	Very high (uniform representation across all entities and data types)
Scalability	Low (compute-intensive for large-scale systems)	High (leverages aggregation properties)	Very high (depends on spatial resolution, temporal resolution, domain size)
Use case examples	Collision avoidance, TCAS, CD&R	Traffic forecasting, airspace complexity assessment	Multi-source integration, location-based risk fields, spatial occupancy, and airspace characterization

Section 2.1.6 identifies DGGS as one of the spatial frameworks used in geospatial modelling and notes its use of the location-based modelling paradigm. However, a comparison of the reviewed airspace-modelling literature shows that trajectory-based and region-based approaches are considerably more prevalent. The identified DGGS-based aviation studies generally apply hierarchical cells to a particular tasks, such as spatial indexing, ADS-B data organization, trajectory representation, conflict searching, path planning, or airspace sectorization. These studies demonstrate the applicability of DGGS to airspace problems but provide limited evidence of its use as the underlying structure of a general airspace model.

A related consideration is integrating heterogeneous airspace data. For example, aircraft surveillance data and weather radar data differ substantially in structure and physical meaning. ADS-B data represent moving aircraft observations sampled along trajectories, whereas weather radar produces spatially distributed environmental observations with varying spatial coverage, sampling intervals, and rates of change. Integrating these sources therefore requires more than storing two datasets within the same system. Their spatial and temporal references must be harmonized so that aircraft and weather conditions can be associated with common locations and times and used to produce joint analytical outputs.

Although geospatial and DGGS frameworks are theoretically well suited to this type of integration, the reviewed airspace literature offers relatively few demonstrations of a common DGGS-based structure for representing multiple heterogeneous airspace entities. Existing DGGS-based aviation applications are generally centred on a single principal data type or a predefined analytical purpose.

Observation #6

Location-based DGGS airspace models are uncommon in the reviewed literature compared with trajectory- and region-based models, and their implementations generally focus on a single principal data type or analytical application rather than integrating heterogeneous airspace entities into a common four-dimensional representation.

2.2.2 Computational efficiency in airspace modelling approaches

Observation #2 in the introduction highlighted that applications using large aerospace datasets must be specifically designed to effectively manage data in ways that optimize computational resources. Two concepts related to optimizing computational resources were identified across the literature on data-driven airspace models: computational power and computational efficiency.

Computational power refers to the raw processing capacity of hardware, such as CPU and GPU resources, used to handle complex calculations, large-scale simulations, and real-time processing [230]. Systems with higher computational power, such as GPU-accelerated architectures and computer clusters, can significantly reduce processing times for air traffic management models, enabling faster responses to dynamic airspace conditions, as is demonstrated by Gianazza et al. [231] for high-fidelity simulations of real-time airspace reconfiguration.

Computational efficiency refers to how well algorithms and models use available computational resources to achieve their objectives. Computational efficiency aims to optimize data processing and modelling techniques to minimize computational load while maintaining accuracy. Efficient algorithms allow for large-scale airspace modelling without requiring excessive computational power. For example, Hickey and Hoffman [232] propose a hybrid queuing-regression model for airspace analysis that provides robust performance estimates without the need for computationally expensive full-scale simulations.

The extent to which airspace models rely on computational power versus computational efficiency varies across modelling approaches. Some models are designed for large-scale analysis and require high-performance computing resources, while others prioritize efficiency through algorithmic optimizations.

Macro-level congestion models, such as system-wide air traffic flow model developed by Xi et al. [80], often require significant computational power due to their reliance on extensive datasets and stochastic traffic evolution simulations. Micro-level models, like the time-dependent Coxian queuing models developed by Wang et al. [233], which focus on individual airport congestion, can be optimized with data-driven predictive analytics, reducing the need for higher computational power.

High-fidelity predictive trajectory models [51] that incorporate uncertainties and probabilistic distributions require high computational power, whereas computationally efficient predictive trajectory models, such as the stochastic linear hybrid system (SLHS) developed by Liu and Hwang [82], utilize neural network-based predictors that process large datasets in real time with reduced computational overhead.

Machine learning-based CD&R models [128] often require substantial computational resources to train deep learning architectures on large historical datasets. In contrast, geometric and rule-based CD&R models rely on deterministic algorithms that are computationally efficient but may be less adaptable to dynamic air traffic environments [124].

Network-based flow control models, like Menon et al.'s Eulerian air traffic flow model [137], which apply graph-based optimization, are scalable and computationally efficient, making them suitable for real-time applications. However, integrating real-time airspace constraints and weather impacts significantly increases computational requirements, necessitating high-performance computing solutions in large-scale airspace management, as demonstrated by Alharbi et al. [53] in their Uncrewed Aircraft Traffic Management prediction model.

Due to their broad scope, system-wide airspace simulation models are among the most computationally intensive airspace model types. For example, McGovern and Kalish [156] developed stochastic airspace simulations using Monte Carlo approaches to assess the effects of uncertainty in airspace operations, requiring large-scale computational infrastructure. Hybrid simulation architectures, like the one developed by Mitchell et al. [217], have been proposed to reduce the need for full-scale real-time computations and to enhance computational efficiency in system-wide airspace simulation models.

Computational efficiency is a key advantage of DGGS-based geospatial airspace models, like Han et al.'s [234] GeoSOT-3D global subdivision grid model, as they allow for scalable resolution and adaptive spatial partitioning. However, when integrating real-time environmental datasets, such as weather and terrain models, computational demands increase.

Unconventional aircraft airspace models like the ones used for Urban Air Mobility (UAM), drones, and hybrid-electric aircraft rely heavily on synthetic data and simulation-based validation due to limited historical data availability. Wang et al.'s study [235] shows that agent-based models that simulate UAM operations in urban environments require high computational power for large-scale traffic simulations. Simplified analytical models like Panayotov et al.'s [166] GIS-based octree model approximates UAM traffic behaviour and provides computationally efficient alternatives but may lack the necessary fidelity for real-world deployment.

High-resolution weather airspace models, like Pensado et al.'s [236] Dynamic Graph model, are used for weather forecasting and incorporate Computational Fluid Dynamics (CFD) simulations that can require significant computational power. Machine learning-based weather airspace models like Wunderli et al.'s

[237] aircraft noise simulation model sonAIR are used for weather predictions and provide computationally efficient alternatives by leveraging historical meteorological data to forecast short-term atmospheric conditions without extensive numerical computations.

The reviewed literature shows that some airspace models are explicitly designed to maximize computational efficiency, while others require high-performance computational power and resources due to their reliance on large datasets and real-time processing. The trade-off between these factors depends on the intended application, with real-time trajectory prediction models often requiring more computational power than strategic air traffic flow simulations.

Observation #7

Different airspace models utilize varying levels of computational power and efficiency.

An important observation from the literature is that optimizing computational efficiency through algorithms and data processing techniques has been shown to yield more scalable outcomes than simply increasing computational power. Wei et al. [238] demonstrated that an optimization-based traffic flow management approach could achieve comparable accuracy to brute-force simulation-based methods while requiring significantly fewer computational resources. Calo et al. [239] emphasize that improvements in algorithmic strategies, such as domain decomposition and sparse direct methods, significantly enhance scalability without relying on excessive computational power. Rajan et al. [240] show that parallelization, distributed computing, and optimized memory management can improve efficiency and allow complex computations to run effectively for a fixed hardware setup. In high-performance computing, Lin and Chan [241] highlight that hierarchical algorithms, such as multigrid and fast multipole methods, have resulted in large computational time reductions, reinforcing the idea that efficient computation is more sustainable than simply scaling up hardware. Schryen [242] explains that computational parallelization has speedup limitations, and scaling computational power alone is often inefficient without corresponding algorithmic optimizations. Collectively, these studies indicate that prioritizing computational efficiency provides more scalable and cost-effective solutions compared to solely increasing computational power. While both factors contribute to high-performance computing, efficient algorithms and processing techniques maximize the potential of fixed hardware, reducing infrastructure costs and improving long-term sustainability in large-scale computational applications.

Observation #8

Prioritizing computational efficiency provides more scalable and cost-effective solutions compared to solely increasing computational power.

Smart feature selection and *dataset partitioning* are examples of two techniques that can be used to increase algorithm computational efficiency and reduce high computational power.

Smart feature selection is a technique that is used to determine which smaller subset of an original dataset can be used for an intended application. More specifically, this approach removes variables which are poorly correlated to the rest of the variables in the original dataset, removing redundant features, or selecting a smaller subset of variables that have a stronger influence on the intended results for the application [243], [244], [245], [246], [247], [248], [249], [250]. Liu and Yue [246] propose dimensionality reduction techniques that help select only the most relevant features, reducing computational overhead without

compromising model accuracy. They argue that eliminating poorly correlated or redundant features significantly improves efficiency and model performance. Manikandan and Kalpana [243] compared feature selection (choosing a minimal number of key variables) vs. feature extraction (transforming data into a lower-dimensional form). Their study concluded that reducing the number of variables is crucial for handling high-dimensional data efficiently. Patidar and Tiwari [248] explored how feature selection eliminates irrelevant and redundant data, making models more interpretable while reducing computational burden. Their research highlights that swarm intelligence algorithms (such as Particle Swarm Optimization and Gray Wolf Optimization) are highly effective for selecting minimal, high-impact variables. Their study concludes that smart variable selection significantly enhances the efficiency of big data models. Kalayci et al. [247] propose using genetic algorithms (GA) for feature selection, demonstrating that GA-based selection methods outperform brute-force approaches in reducing dimensionality. Their findings confirm that using a smaller, optimized set of features speeds up big data processing while maintaining model accuracy.

Dataset partitioning is a technique used to subdivide a larger original dataset into smaller more manageable parts (called chunks) to perform computationally expensive operations on it (as opposed to the larger original dataset). The technique typically keeps track of the relationship between each chunk so that it can recombine the results related to each chunk into one larger dataset representative of the original dataset. For example, a spatiotemporal dataset can be partitioned into chunks using time intervals and computationally expensive operations like 3D distance calculations can be performed on this smaller dataset rather than on the original dataset. Ramkumar et al. [251] explains that breaking large-scale computational problems into smaller, manageable chunks allows for faster processing while maintaining efficiency. They argue that chunking enables optimization by balancing computational cost and acceptable results. Yang and Chen [252] propose a data partitioning approach where big data is divided into chunks, enabling selective compression and improving processing speed while minimizing storage overhead. Xia et al. [253] present a parallelized approach to data chunking, where large datasets are processed in separate threads and reassembled later to reduce computational bottlenecks while maintaining data accuracy. Misailovic [254] discusses the balance between computational efficiency and the accuracy of results. Their research highlights that not all big data applications require extensive computational power, as optimizing algorithms for performance can often yield acceptable results with significantly reduced processing costs. Ali et al. [255] discuss energy-efficient big data networks, emphasizing that progressively processing data in strategic locations reduces computational load, minimizing the need for excessive power consumption.

2.2.3 Multi-vehicle type support in airspace models

A trend observed in the literature is that most airspace models are designed to simulate the behaviour of a single type of air vehicle, such as fixed-wing aircraft or drones, and relatively few models incorporate multiple vehicle types. Multi-vehicle modelling is an important feature for modern airspace modelling, particularly with the increasing presence of diverse and unconventional aircraft types such as urban air mobility (UAM) vehicles, eVTOLs, and hybrid-electric aircraft.

Several airspace models have attempted to address multi-vehicle integration challenges. System-wide airspace simulation models, such as those proposed by Thipphavong et al. [256], incorporate mixed air traffic, evaluating how tailored trajectory management (TTM) capabilities impact both crewed and uncrewed regional air cargo operations. The study demonstrated that pre-emptive conflict resolution strategies can reduce the likelihood of UAV-aircraft conflicts and improve airspace safety.

Geospatial airspace models use structured spatial representations to integrate heterogeneous aircraft types into a common framework. Alharbi et al. [53] demonstrated that deep learning-based air traffic prediction

models can be trained on heterogeneous datasets, incorporating data from traditional commercial aviation and emerging UAS operations.

Congestion prediction models typically focus on either conventional or unconventional aircraft, rather than integrating both simultaneously. However, Tian and Hu [257] developed a multi-objective genetic algorithm that allows real-time congestion prediction across different airspace users, demonstrating that multi-vehicle congestion forecasting is feasible with optimization-based approaches.

Predictive trajectory models generally focus on either commercial aviation or UAV operations but rarely both. Cheng et al. [258] introduced a machine learning-aided trajectory prediction model that accounts for both crewed and uncrewed aircraft operations. Their research highlights the need for more robust trajectory synchronization mechanisms when dealing with highly dynamic airspace environments.

Traditional geometric and rule-based CD&R models often assume homogeneous air traffic. Liu et al. [58] developed a network-based CD&R approach designed specifically for dense mixed-traffic airspace with both crewed and uncrewed aircraft. Their results show that centralized conflict resolution strategies can be adapted to mixed-traffic environments, but further improvements are needed to reduce computational complexity. Ho et al. [57] proposed a multi-agent-based CD&R method for UAV operations in shared airspace, demonstrating that scalability is a major challenge when modelling both conventional and unconventional vehicles in high-density environments.

Most flow control models focus on single-vehicle types. Menon et al. [133] developed a predictive control model for air traffic flow management, which primarily considers conventional aircraft. However, Gharibi [259] introduced a multi-vehicle UAS flow control model, proposing a layered Internet of Drones (IoD) architecture that assigns velocities to UAVs dynamically based on traffic density. Their research demonstrated a need for flow control methodologies to evolve to incorporate new entrants into existing airspace structures.

Weather models play a key role in optimizing air traffic routes but traditionally focus on commercial aviation. However, recent research has emphasized the need for weather-aware trajectory planning for UAV and eVTOL operations. Fernandes et al. [260] demonstrated that trajectory synchronization across different aircraft types could enhance route efficiency, particularly in the presence of weather-related constraints.

These studies show that most airspace models are designed for a single type of air vehicle, with only a few efforts incorporating multiple vehicle types. Congestion prediction models, predictive trajectory models, conflict detection and resolution models, geospatial airspace models, and system-wide airspace simulation models have shown promise in handling heterogeneous aircraft operations. However, scalability and computational efficiency remain significant challenges when integrating conventional aircraft with new entrants like UAVs, eVTOLs, and hybrid-electric aircraft.

Observation #9

Most airspace models are designed for a single type of air vehicle, with only a few efforts incorporating multiple vehicle types.

As emerging airspace technologies continue to evolve, the ability of airspace models to integrate new entrants, such as Urban Air Mobility (UAM), Electric Vertical Takeoff and Landing (eVTOL) vehicles, drones, and hybrid-electric aircraft, alongside conventional aircraft (airplanes and helicopters) has become an important research focus. However, existing models largely fall into three distinct categories: those that

only model new entrants, those that only model conventional aircraft, and those that incorporate both new entrants and conventional aircraft into a single framework.

Several airspace modelling studies focus exclusively on new entrants without integrating conventional aircraft operations. For instance, Panchal et al. [261] developed a collision detection and avoidance model designed for UAM operations, addressing conflict management challenges unique to UAM vehicles, including non-cooperative airspace users such as birds and smaller drones. The model does not incorporate conventional air traffic, limiting its applicability in integrated airspace operations. Cotton [72] proposed an adaptive autonomous separation approach for UAM operations, focusing on urban and short-range intercity transportation while excluding interactions with traditional piloted aircraft.

Conversely, many legacy airspace models are designed for conventional aircraft only, without considering operational complexities introduced by new air vehicle types. Chaturmutha and Mathew [262] discussed airspace modelling limitations in integrating unmanned aircraft (UAS) and space transportation operations within existing air traffic management systems. These legacy models primarily rely on traditional air traffic control paradigms optimized for fixed-wing commercial aircraft and rotorcraft, making integration with new entrants difficult.

A smaller number of studies attempt to model both new entrants and conventional aircraft within the same framework. Anaroua and Babiceanu [263] introduced a risk analysis-based approach for integrating UAS operations into controlled and uncontrolled airspace, highlighting the need for harmonized airspace management strategies that address both conventional and new aircraft types. Additionally, Andreeva-Mori [264] proposed trajectory-based airspace modelling techniques that define safety buffers and geo-fencing mechanisms to accommodate both conventional and advanced air mobility (AAM) operations.

The literature shows that while some models integrating both conventional and unconventional aircraft exist, most airspace models are tailored to either conventional aircraft or new entrants, but not both. While a few frameworks support mixed operations, further research is required to develop scalable, integrated airspace models that accommodate both existing aviation operations and future air vehicle technologies.

Observation #10

The majority of airspace models are tailored to either conventional or unconventional aircraft, but not both.

2.2.4 Differences between real-time, historical and simulated data

The integration of unconventional aircraft into national airspace presents several modelling challenges. One of the primary difficulties is the availability and reliability of data sources, which fall into three broad categories: real-time data, historical data, and simulated data. Each of these data types has unique characteristics, applications, and limitations with respect to their use in airspace modelling.

2.2.4.1 Real-time data in airspace modelling

Real-time data is used for time-sensitive decision-making in airspace operations, particularly for ATM, conflict detection, and trajectory prediction. Hopkin [265] discusses how real-time simulation techniques have been applied in air traffic control (ATC) to support procedural training, airspace safety, and dynamic rerouting. However, one of the challenges of real-time data is its large volume and velocity, requiring high computational efficiency and power to process and integrate into airspace models effectively.

For unconventional aircraft, Schmitt et al. [266] introduced a real-time simulation framework for UAV integration into non-segregated airspace. Their study highlights how real-time data collected from UAVs operating under ATC control can provide insights into controller workload, separation requirements, and operational feasibility. Hochrainer and Schattovich [267] demonstrated how hybrid real-time simulation approaches can be used to optimize UAV electronic control systems, closing the loop between physical UAV hardware and computational models.

While real-time data is valuable, its reliability can be affected by sensor malfunctions, communication latency, and data transmission errors, necessitating robust validation techniques [268].

2.2.4.2 Historical data in airspace modelling

Historical data serves as an important reference for modelling airspace operations, particularly for analyzing long-term trends, traffic patterns, and system-wide efficiency. Zelinski [269] discusses how historical air traffic records, flight plans, and operational data have been used to validate airspace concept evaluation systems, ensuring realism in congestion prediction and traffic flow models.

However, there is a fundamental lack of historical data for new entrants, making it difficult to develop predictive models. Shresta et al. [105] addressed this limitation by identifying UAS traffic from historical radar data to estimate the impact of UAV operations in controlled airspace. Their study demonstrated how historical air traffic data can be leveraged to generate hypothetical flight plans for new entrants, offering a way to simulate future integration scenarios.

Despite its utility, historical data is often biased toward conventional airspace operations and may not fully represent the behaviour of emerging technologies, necessitating alternative data-generation techniques.

2.2.4.3 Simulated, synthetic, and hybrid data in airspace modelling

A third type of data found in the airspace modelling literature is data derived from historical real-world data, which includes *simulated*, *synthetic*, and *hybrid* data types. This type of data is prevalent in unconventional aircraft (UAS, UAM, eVTOL, etc.) research, where historical real-world data is often lacking due to their novelty.

Simulated data is derived from historical data using either physics-based or system-level models to explore specific scenarios and evaluate system behaviour. For instance, Ferrare et al. [270] developed a multi-agent simulation framework in NetLogo to model UAM operations and assess separation and congestion risks. Azad et al. [271] introduced the A2A-SDD dataset to simulate drone interactions and to train AI models for collision avoidance and trajectory optimization. These simulated environments enable rapid prototyping but depend heavily on assumptions about future operations and flight behaviours.

Synthetic data, in contrast, is artificially generated to match the statistical properties of real-world data, often using generative machine learning methods, rather than obtained through direct measurement or real-world observation. It is often used when real data is scarce, sensitive, or unavailable due to privacy, security, or classification. Dehnavi et al. [272] emphasized the role of synthetic data in privacy-preserving model development and highlighted its scalability for AI applications in aviation. Kochenderfer et al. [204] developed probabilistic encounter models to generate synthetic datasets of aircraft encounters for evaluating the development of Traffic Collision Avoidance Systems (TCAS) and ACAS X. These encounter datasets include thousands of simulated conflict scenarios, spanning diverse encounter geometries and operational conditions and have become foundational in collision avoidance analysis and system certification [273]. Wijnands et al. [274] investigated the applicability of a TimeGAN model in generating synthetic 4-

dimensional aircraft landing trajectories that preserve traffic dynamics and common traffic patterns to help analyze airspace constraints and delay propagation.

Hybrid approaches that combine simulated or synthetic data with historical or real-time operational data also exist. Souanef et al. [275] proposed a digital twin framework for UTM that integrates both simulated drone operations and real-world flight telemetry to validate airspace procedures. Borener et al. [205] used a combination of radar tracks and stochastic simulations to assess separation risk for UAS operations in metroplex environments, showing that blended data sources improve reliability.

Both simulated and synthetic data introduce inherent uncertainties in model validation. Consequently, hybridization offers a promising path forward, providing a bridge between theoretical projections and real-world behaviour.

Observation #11

Real-time and historical data are widely used in airspace models for conventional aircraft.

Observation #12

Real-time and historical data for unconventional aircraft are largely unavailable and researchers frequently rely on simulated or synthetic data in unconventional aircraft airspace models.

2.2.5 Importance of full 3D analysis in airspace models

Airspace models vary in their level of spatial complexity. Some models use simplified 2D representations for computational efficiency, while others employ full 3D models to more accurately represent real-world airspace conditions.

Most congestion prediction models rely on macro-level system-wide modelling to identify air traffic bottlenecks. Traditionally, 2D sector-based models have been used to assess congestion based on en-route airspace demand. However, modern congestion models are evolving toward 3D airspace partitioning for better traffic flow predictions. For instance, Huang et al. [276] introduce a three-dimensional metacell transport model, which refines congestion prediction by dividing the airspace into 3D cells that track vertical and horizontal aircraft movement patterns.

Predictive trajectory models require high-resolution spatial modelling to accurately forecast future aircraft positions. Many older trajectory models rely on simplified 2D path projections, assuming minimal altitude variations. However, Ayhan and Samet [51] developed a stochastic trajectory prediction model based on a 3D grid network, improving trajectory predictions by incorporating spatial variations in weather and air traffic conditions.

Many CD&R models have historically relied on 2D conflict detection algorithms, which assume constant altitude bands for separation assurance. However, Han et al. [277] propose an optimized 3D conflict resolution approach that divides airspace into digitized 3D grids, improving multi-aircraft conflict resolution under varying airspace constraints. Gianazza and Durand [278] also highlight how 3D-separation algorithms provide more accurate conflict detection by considering vertical deviation possibilities beyond 2D lateral conflict resolution techniques.

System-wide airspace models must be scalable to represent large airspace sectors, often prioritizing computational efficiency over full 3D spatial fidelity. Some simulation models simplify vertical separations into predefined flight levels, avoiding fully dynamic altitude modelling. However, Mallick et al. [279] developed a 3D aircraft trajectory estimation model that integrates multiple radar data sources, allowing for a more realistic representation of aircraft motion in complex airspace environments.

Geospatial airspace models use GIS-based techniques to analyze airspace structures, air traffic flow, and environmental constraints. Traditionally, these models have been 2D-focused, mapping air corridors on flat projections. However, the use of Discrete Global Grid Systems (DGGS) in geospatial airspace models enables full 3D spatial representation. Baier and Zimmer [280] demonstrate how 3D GIS-based conflict detection improves air traffic safety by providing accurate real-world spatial mapping.

Many airspace models still use 2D sector-based representations, where altitude is treated as a set of discrete layers rather than a continuous third dimension. While this simplification reduces computational complexity, it can lead to inaccurate conflict detection and suboptimal trajectory predictions. For example, Gianazza and Durand [278] show that 3D-separation algorithms allow for better air traffic flow management by optimizing aircraft altitude assignments dynamically rather than relying on fixed layers. Han et al. [277] introduce a digitized 3D grid-based conflict resolution system, which significantly improves multi-aircraft deconfliction compared to 2D approaches. 3D modelling enhances air traffic situational awareness, improves collision avoidance, and allows for optimized flight trajectories. For example, Ayhan and Samet [51] propose a stochastic trajectory prediction model based on a 3D airspace grid, demonstrating that 3D spatial structuring improves the accuracy of flight trajectory forecasting.

Observation #13

Both layered two-dimensional and fully three-dimensional representations are used in airspace modelling; comparative studies report improved conflict, separation, and trajectory analyses when altitude is represented explicitly.

2.2.6 Importance of time-varying analysis in airspace models

Airspace models differ in their approach to temporal complexity. Some models operate under static assumptions, maintaining fixed airspace structures and traffic distributions, while others incorporate dynamic variables to better reflect real-world conditions. Time-dependent models account for variations in traffic density, aircraft trajectory adjustments, and evolving weather conditions, making them more representative of operational airspace behaviour.

Early congestion prediction models often assumed static airspace structures, limiting their ability to capture evolving traffic dynamics. More recent approaches integrate time-adaptive air traffic density to enhance congestion forecasting. For example, Tian et al. [281] developed a probabilistic en-route sector traffic demand prediction model by estimating distributions of sector entry, exit, and transit times; enabling sector congestion risk forecasts over future intervals and demonstrating superior accuracy compared to static models. Amor and Bui [56] introduced a multi-agent congestion dynamics model that simulates time-dependent traffic variations within European airspace, showing the impact of evolving demand on sector congestion.

Traditional trajectory prediction models were largely deterministic, assuming fixed flight paths and static environmental conditions. However, recent studies incorporate stochastic elements to account for time-dependent uncertainties. Matsuno and Tsuchiya [282] proposed a stochastic 4D trajectory optimization

approach that integrates real-time weather updates and traffic fluctuations to refine trajectory accuracy. Ayhan and Samet [51] leveraged machine learning techniques to develop a predictive trajectory model that adjusts based on dynamic environmental factors, improving situational awareness and airspace flow management.

CD&R models must incorporate time-dependent variables to provide accurate and reliable predictions. Jilkov et al. [52] introduced a multiple-model approach for CD&R, incorporating aircraft intent and weather uncertainty into conflict prediction. Their method dynamically adjusts conflict detection sensitivity based on real-time conditions, improving the safety and efficiency of conflict resolution strategies. Mazzotta [283] developed a probabilistic CD&R framework that factors in variable weather hazards and sensing uncertainties, enhancing the robustness of conflict resolution methods.

Airspace models designed to study unconventional aircraft, such as UAS and UAM, require time-sensitive modelling for highly dynamic flight operations at lower altitudes. Wu et al. [284] proposed a trajectory-based flight scheduling framework for UAM, integrating real-time congestion adaptation to optimize airspace utilization. The study highlights the necessity of dynamic scheduling techniques to accommodate the unpredictability of low-altitude air traffic. Rodionova [285] investigated air traffic optimization in oceanic airspace, demonstrating how temporal adaptation is important for strategic routing in long-haul flights.

Observation #14

Airspace models differ in their approach to temporal analysis, with some models relying on static representations while others integrate dynamic time-dependent variables.

Weather models are characterized by differences in temporal scales when compared to aircraft movements. Aircraft require high-frequency updates due to their rapid changes in position, whereas weather phenomena, such as convective systems and pressure changes, progress at a slower pace and can be updated less frequently. This discrepancy in time scales poses challenges to integrating different airspace elements into a single model.

Some models from the literature address this discrepancy by implementing multi-resolution approaches. For instance, Henderson [286] explores a stochastic demand and capacity model that integrates weather and air traffic control uncertainty, demonstrating that incorporating time-dependent traffic constraints improves congestion prediction while accounting for different update frequencies. Roy et al. [158] propose a stochastic network model for weather-impact evolution at different time horizons, illustrating how multi-resolution models can accommodate both short-term and long-term weather variations while providing useful decision support for traffic flow management.

Many existing airspace models, however, are tailored to a specific timescale focused on the rapid dynamics of aircraft or the slower evolution of weather systems, but not both. McGovern and Kalish [156] developed a stochastic airspace simulation tool that captures aircraft kinematics with high precision, but their approach does not fully incorporate the slower-evolving impacts of weather changes over extended periods. Pensado et al. [236] explore high-resolution atmospheric prediction models for Advanced Air Mobility (AAM), emphasizing the importance of real-time Computational Fluid Dynamics (CFD) simulations for weather forecasting. Their findings indicate that while CFD-based models effectively capture turbulence and microclimatic variations, their high computational demands make them unsuitable for real-time integration with fast-moving air traffic models.

The challenge of integrating entities with disparate rates of change remains an area of ongoing research. Some recent studies attempt to bridge this gap using hybrid models. Clarke et al. [194] aimed to integrate entities with disparate rates of change in their model by proposing a stochastic airspace capacity model that combines ensemble weather forecasts with air traffic control algorithms, enabling real-time updates that balance aircraft maneuverability with evolving weather conditions.

Observation #15

Aircraft, weather systems, and other airspace entities evolve at different temporal resolutions, complicating their integration within a common time-varying model.

Observation #16

Most airspace models are designed to model airspace elements (aircraft, weather, etc.) using either high-frequency or low-frequency time-scale processes, with relatively few efforts focused on merging multiple time scales in one same model.

A part of the literature on airspace models acknowledges that failing to incorporate dynamic temporal components may lead to inaccuracies in real-world applications. Srivastava [287] developed a rapid assessment model for air traffic impact analysis using a "what-if" paradigm. The study found that models incorporating time-varying conditions, such as airspace blockages and changing traffic patterns, performed better than static models. Their results showed that neglecting time-varying factors could lead to inaccuracies in delay estimation and conflict detection. Tian and Hu [257] introduced a probabilistic airspace sector traffic demand prediction model, which accounts for variations in aircraft entry and exit times. Compared to traditional deterministic approaches, the model reduced unnecessary flow restrictions by 30 minutes and improved air traffic controller workload efficiency. This study highlights that static models often fail to anticipate real-time congestion fluctuations. Sui et al. [288] designed a dynamic air traffic situation prediction model using a spatiotemporal graph convolutional network (STGCN). Their model incorporated real-time operational data to predict multi-sector air traffic flow changes. Experimental results demonstrated that time-sensitive models outperformed traditional static models, achieving over 90% prediction accuracy. Lauderdale [289] developed Bayesian network-based predictive models for the NAS, which use only historical data for congestion prediction. While the models improved congestion awareness, they were unable to dynamically adjust to real-time changes in air traffic, limiting their effectiveness in real-world applications. Amor and Bui [56] proposed a discrete cellular automata and multi-agent system for modelling airspace congestion dynamics. Their model highlighted the existence of a traffic threshold that leads to airspace saturation. The study emphasized that models relying on static configurations often fail to capture phase transition phenomena, making them less reliable under high-traffic scenarios.

Observation #17

Models that fail to incorporate time varying airspace elements are less accurate for real-world applications when compared to the models that do.

2.2.7 Model adaptability to different geographical regions

Most airspace models are developed to focus on either terminal/local airspace (airport approach and departure areas) or National Airspace System (NAS)-wide regions (en-route airspace and system-wide traffic flow).

Many airspace models are designed specifically for terminal area operations, which require high-resolution modelling of aircraft trajectories, separation procedures, and runway interactions. Vempati et al. [290] present an approach to modelling terminal area trajectories within the National Airspace System Performance Analysis Capability (NASPAC); a simulation and analysis framework used by the FAA and related researchers to evaluate how the U.S. National Airspace System performs under different traffic, delay, capacity, and operational scenarios; highlighting the complexities associated with terminal airspace, including detailed arrival and departure procedures, merging and sequencing algorithms, and high-density operations near airports. The study emphasizes that modelling terminal areas requires a different level of granularity than NAS-wide simulations, as individual aircraft movements must be simulated with high precision. Tandale et al. [230] introduce two simulation tools, Queue Generator (Q-Gen) and Discrete Event Queuing Simulator (DEQS), for analyzing congestion in major terminal areas. These models focus on arrival sequencing, aircraft separation at terminal entry gates, and airport delays, illustrating the unique challenges of modelling terminal airspace independently from en-route airspace.

Conversely, NAS-wide airspace models are typically designed to capture high-level air traffic flows rather than detailed operations in specific locations. Odoni and Simpson [106] provide a review of NAS models, categorizing them by their ability to assess capacity, delays, controller workload, and air route traffic management at the system-wide level. The study highlights that most NAS-wide models are not well-suited for detailed airport-level modelling, as they focus on large-scale airspace structures rather than individual aircraft interactions. Shen [291] developed an agent-based NAS-wide model to analyze airline and passenger dynamics, emphasizing airlines' strategic decision-making, delays across the entire NAS, and interactions among major airport hubs. This large-scale modelling approach contrasts with terminal models, which require finer spatial and temporal resolution.

Observation #18

There exist two main approaches in the literature for modelling the airspace for different geographical regions: terminal/local airspace (airport approach and departure areas) and National Airspace System (NAS)-wide regions (en-route airspace and system-wide traffic flow).

The need for these two distinct modelling approaches arises from differences in operational focus and data requirements. Terminal airspace models require detailed information on aircraft sequencing, taxiing, and runway utilization, making high-resolution tracking necessary. In contrast, NAS-wide models prioritize traffic flow management, route optimization, and large-scale capacity planning, where broad spatial coverage is more important than individual aircraft details.

Observation #19

The two modelling approaches differ in spatial resolution and level of detail. Terminal models use high-resolution geographical scales for precise aircraft interactions and runway operations, whereas NAS-wide models make use of large-scale geographical scales with lower resolution for air traffic flow and delay analysis.

The reviewed literature shows that most airspace models are specialized for either terminal/local airspace or NAS-wide airspace, but not both. Although much fewer, there have been efforts to integrate both terminal and NAS-wide airspace within a single framework [292], [293], [294]. Couluris et al. [292] discuss the Airspace Concepts Evaluation System (ACES), a multi-scale simulation tool that incorporates both terminal and en-route operations. The ACES model includes airport, terminal, en-route, and system-wide traffic management agents, providing a more comprehensive representation of the NAS. The authors acknowledge that maintaining modelling fidelity across different spatial scales remains a challenge, as high-resolution terminal modelling can significantly increase computational complexity in a NAS-wide framework. Long et al. [294] develop an integrated air traffic management model under NASA's Terminal Area Productivity (TAP) and Advanced Air Transport Technologies (AATT) programs. Their queuing network model adjusts for different airspace sectors, demonstrating how NAS-wide and terminal operations can be analyzed together.

Some of these airspace models [213], [295], [296], [297] combine terminal/local airspace or NAS-wide airspace in one same model based on the principle of using a variable geographical resolution approach. Holdren et al. [295] developed a pipeline that constructs flexible graph-structured data from flight trajectories for air traffic analysis. Their approach allows adaptation to different resolution levels depending on the use case, demonstrating that higher resolution is required for detailed analyses while coarser resolution is sufficient for broader traffic flow modelling. Their study highlights the importance of scalable, multi-scale airspace modelling to accommodate different levels of granularity efficiently. Gurtner et al. [296] applied multi-scale network community detection to the European airspace, analyzing different traffic layers at various resolutions. Their research highlights the need for adaptable resolution models that can transition from detailed local-level representations to large-scale en-route airspace networks.

Observation #20

There have been limited efforts to integrate both terminal and NAS-wide airspace within a single framework.

2.2.8 Model visualization

As airspace models increase in complexity, the need for robust visualization tools has grown. Modern airspace models must integrate large amounts of data, including three-dimensional spatial information; temporal dynamics; and diverse datasets such as weather patterns, traffic congestion, and vehicle performance characteristics. Without effective visualization techniques, analyzing and interpreting these models becomes cumbersome, limiting their practical usability.

3D visualization tools are widely recognized as essential for comprehending spatiotemporal air traffic patterns. Wang and Xu [298] developed a tool specifically for 3D visualization of moving objects, highlighting the benefits of graphical representation over traditional numerical data. Their approach

demonstrated how dynamic animation and real-time tracking provide deeper insights into air traffic movement than simple tabular or 2D representations. Debiasi et al. [299] emphasize the role of interactive visualization in analyzing large-scale airspace operations. Their study on urban traffic simulation suggests that conventional visualization interfaces often have a steep learning curve and fail to effectively communicate simulation results. They propose an intuitive 3D visual analytics tool to enable real-time exploration of air traffic flows, reducing cognitive overload for analysts and decision-makers. David and Tauro [300] developed a web-based 3D visualization framework that allows users to intuitively navigate and analyze large spatiotemporal datasets. Their study argues that interactive 3D representations significantly improve analytical usability, enabling air traffic controllers and researchers to assess evolving traffic patterns efficiently. Crowell et al. [301] developed FliteViz4D, an interactive 4D visualization system for air traffic analysis. This tool extends traditional 3D visualization by incorporating time as an additional dimension, enabling analysts to visualize both present and predicted traffic conditions simultaneously. The study underscores that 4D visualization is particularly beneficial for concepts such as Trajectory-Based Operations (TBO) and Path Arrival Management, where real-time decision-making is required. Cui. [302] developed a data visualization tool that includes an interactive 4D map that is used to enhance data understanding found inside airspace environments. The tool is tested through a case study focusing on air traffic interactions in a shared airspace, leading to the creation of a relational database with archived air traffic and weather data. The result is a flexible approach for creating realistic flight scenarios and assessing risk, demonstrating the tool's utility.

Observation #21

3D and 4D visualization tools play an important role in the interpretation of airspace models.

Recent advancements in both aerospace research and data science [303], [304], [305], [306], [307], [308] provide strong support for leveraging open-source tools—particularly Python libraries (modular blocks of pre-built Python code with specific functions)—for high-dimensional and spatiotemporal data visualization. Some examples of this include Python-based visualization libraries such as Psyplot, Cartopy, and Matplotlib. Sommer [309] shows how Psyplot leverages the xarray and Cartopy libraries to allow the visualization of scalar and vector fields across space and time. Using these Python libraries to not only build the airspace model but also its visualizations avoids the need for building specialized tools from scratch and promotes a modular, scalable approach to airspace model analysis all within a single open-source Python ecosystem. These Python-based libraries are also compatible with other geospatial data structures (DGGS for example) and with commonly used clustering and machine learning libraries (Scikit-learn, HDBSCAN, and UMA).

2.2.9 Adaptability to multiple applications

Most airspace models are designed for specific conditions and are not inherently adaptable to be reusable across a wide range of applications.

Airspace models are generally designed to focus on a specific application, such as congestion prediction, trajectory forecasting, conflict detection, midair collision risk analysis or air traffic flow management. Many models lack built-in adaptability, requiring substantial modifications when applied to new applications. Examples include the dynamic airspace-configuration models developed by Sergeeva et al. [310] and Savai et al. [311], the trajectory-flexibility model developed by Hasan and Rossiter [312], the terminal-sector design model developed by Sciandra et al. [313], and the operational risk-assessment model

presented by Mehadhebi [314]. Each model was developed around a particular analytical objective, and the reviewed studies do not demonstrate reuse of the same core representation across these distinct applications.

According to Durak [315], the primary challenge in reusing airspace models stems from interoperability and composability issues, where models are often built for distinct operational paradigms and cannot easily integrate with other frameworks. Durak explains that assumptions and constraints are imposed in airspace models which can limit their reusability.

Models designed for congestion prediction applications, for example, often employ sector-based optimization strategies to manage controller workload and balance air traffic flows. These models, while effective in their intended application, struggle to accommodate trajectory prediction or collision risk assessment without extensive adjustments [310]. Similarly, models focused on trajectory prediction prioritize flexible path forecasting methods but, according to Hassan and Rossiter [312], may lack the built-in structure needed for congestion modelling or airspace safety assessments.

Efforts to develop adaptable models have emerged, but their scope remains limited. Dynamic Airspace Configuration (DAC) models [311] introduce adaptable sector designs that adjust to fluctuating traffic patterns. While DAC models [311] have shown success in improving flexibility for congestion management, they have proven less effective when adapted for collision risk prediction or trajectory forecasting without extensive reconfiguration.

Another example is terminal airspace optimization models, which focus on dynamically adjusting sector boundaries in response to varying airport demand and traffic complexity. While effective for local airspace congestion, these models are often unable to efficiently adapt for system-wide collision risk or broader trajectory analysis [313].

Additionally, risk assessment models like the Collision Risk Model (CRM) [314] are highly specialized for identifying operational hazards but require significant modifications to handle broader airspace applications such as flow control or dynamic sectorization.

The distinction between data integration and analytical reuse is important. A model may incorporate several data sources while remaining designed around one predefined application. Conversely, a reusable airspace model separates the common representation of the airspace from the analytical methods applied to that representation. New metrics or analytical methods can then operate on the same model products without requiring the underlying airspace structure to be redesigned.

This limitation is also evident in the reviewed DGGS-based aviation applications. Although these studies demonstrate the suitability of DGGS for specific functions such as data indexing, trajectory management, spatial searching, conflict detection, or sectorization, they generally do not evaluate whether the same DGGS-based airspace representation can support multiple distinct analytical applications. Consequently, the use of DGGS alone does not ensure model reusability; reusability also requires modular separation between the representation of the airspace and the application-specific methods that operate on it.

Overall, while individual models have demonstrated success in targeted domains, the absence of modular and flexible frameworks that can efficiently adapt for multiple airspace applications remains a critical challenge. Future research may benefit from integrating modular design principles that allow core modelling frameworks to extend across operational scenarios with minimal reconfiguration.

Observation #22

Most airspace models are designed for specific operational applications and often lack the flexibility to be reused across multiple applications without significant modifications.

2.2.10 Role of subject matter expertise in airspace modelling

A common trend observed in the airspace modelling literature is that some airspace models include assumptions and biases based on subjective expert opinion rather than objective observations from actual data. This reliance on expert judgment, while sometimes necessary due to the lack of complete datasets, can introduce potential errors and lead to inconsistencies in model predictions.

Donohue et al. [316] discuss how many airspace models rely on hidden assumptions embedded within the model code that are often only known to the developers. These assumptions, which may be based on expert knowledge rather than empirical validation, can significantly influence predictions related to traffic flow, delays, controller workload, and safety assessments. The study highlights that few airspace models undergo extensive validation using real-world operational data, making it difficult to assess the accuracy of their assumptions.

The reliance on expert judgment in airspace models is further examined by Burgman et al. [317], who emphasize the impact of cognitive biases such as anchoring, confirmation bias, and overconfidence when eliciting expert input. Their research outlines how expert-driven assumptions can shape probabilistic risk models, which are commonly used in airspace simulations. The study also highlights that structured group elicitation techniques, where multiple experts provide input independently before discussion, can mitigate some of these biases and improve model robustness.

Another example of expert-driven modelling is found in Katz et al. [318], who developed an Urban Air Mobility (UAM) encounter model based on expert judgement rather than historical flight data. Due to the lack of real-world UAM flight trajectories, the study relied on Markov decision process (MDP) modelling tuned using expert-provided scenarios. While this approach allowed for rapid model development, it also introduced subjectivity in terms of assumed separation distances, reaction times, and conflict resolution strategies, which could lead to inconsistencies once real-world data becomes available and even limit the scope of critical scenarios considered in these types of studies.

Smirti and Hansen [319], in their assessment of NASA's Virtual Airspace Modelling and Simulation (VAMS) project, discuss how expert-derived assumptions in airspace planning should be carefully documented and tested for sensitivity. The study found that simulation tools can be valuable for modernization efforts, but the assumptions made—particularly those based on expert intuition without real world validation and testing—should be transparent and validated against empirical data.

Observation #23

Many airspace models incorporate assumptions and constraints reliant upon subject matter expertise.

2.2.11 Summary of each observation for each airspace model

Table 9 compares the nine airspace model families introduced in Section 2.1 against cross-cutting modelling capabilities synthesized from Observations #6 through #24. The symbols summarize the strength of evidence identified across the reviewed implementations in the literature within each model subtype; they do not imply that every model belonging to a subtype, or any single implementation, exhibits all of the marked capabilities. A checkmark (✓) indicates that the capability has been clearly demonstrated in the reviewed literature, a tilde (~) indicates that it has been demonstrated only in limited cases or requires substantial adaptation, a cross (×) indicates that it is generally absent or has not been demonstrated, and a hyphen (–) indicates that the capability is not applicable.

Table 9 indicates that no individual implementation reviewed in the literature has been consistently demonstrated to combine all of the identified capabilities. Probabilistic and machine learning-based predictive-trajectory models exhibit broad capabilities, particularly for three-dimensional, time-varying analysis of aircraft motion and interactions. They have also been applied to different vehicle types and operational settings. However, these capabilities are distributed across different implementations, and adaptation to new geographical regions or analytical applications commonly requires recalibration, retraining, or modification of the trajectory model.

Predictive-trajectory models may incorporate heterogeneous inputs such as aircraft state, weather, flight-plan, and intent information. Nevertheless, these inputs are generally incorporated to improve the prediction or analysis of aircraft trajectories, which remain the primary analytical objects. This differs from a location-based model in which aircraft, weather, and other entities are associated with a common spatial and temporal structure and can contribute to derived values assigned to fixed locations.

DGGS-based geospatial models provide promising structural properties for three-dimensional, time-varying, multi-resolution representation and spatial indexing. Their cell-based structure is particularly well suited to integrating heterogeneous observations and supporting location-based outputs. However, the identified aerospace literature implementations remain comparatively limited and are generally developed for specific datasets or analytical functions. Evidence of heterogeneous multi-data source integration and reuse of the same DGGS-based representation across multiple analytical applications therefore remains limited.

Table 9: Airspace model comparison with respect to each identified observation

Model type	Model sub type	Observations								
		Computational efficiency	Multi-vehicle support	Unconventional aircraft support	3D analysis	Time varying analysis	Adaptability to different airspace region	Model visualization tools	Multiple use cases possible	Heterogenous multi data source integration
Congestion prediction models	Mathematical models and optimization techniques	✓	✓	×	×	✓	✓	×	~	~
	Probabilistic models	×	✓	~	✓	✓	✓	×	~	~
	ML-based models	✓	✓	✓	✓	✓	✓	✓	~	~
Predictive trajectory models	Physics-based models	×	×	×	✓	✓	✓	~	~	~
	Probabilistic models	~	✓	✓	✓	✓	~	✓	~	~
	ML-based models	~	✓	✓	✓	✓	~	✓	~	~
Conflict detection & resolution (CD&R) models	Geometric models	✓	×	×	✓	✓	×	×	~	×
	Rule-based models	✓	✓	✓	✓	✓	✓	✓	~	~
	Probabilistic models	✓	✓	✓	✓	✓	✓	✓	~	~
	ML-based models	✓	✓	✓	✓	✓	✓	✓	~	~
Flow control models	Mathematical models	✓	×	×	×	✓	✓	×	~	~
	Optimization techniques	✓	✓	~	✓	✓	✓	✓	~	~
	ML-based models	✓	✓	✓	✓	✓	✓	✓	~	~
System-wide simulation models	Agent-based models	×	✓	×	✓	✓	✓	✓	~	✓
	Stochastic models	×	✓	✓	✓	✓	✓	✓	~	✓
	Hybrid models	×	✓	✓	✓	✓	✓	✓	~	✓
Geospatial airspace models	GIS-based spatial models	✓	✓	~	✓	~	~	✓	~	✓
	DGGS-based models	✓	✓	✓	✓	✓	✓	✓	~	~
Unconventional aircraft models	Simulation-based models	✓	✓	✓	✓	✓	×	✓	~	✓
	Optimization-based models	✓	✓	~	✓	✓	×	✓	~	~
	ML-based models	✓	✓	✓	✓	✓	×	✓	~	~
Airspace weather models	Regression models	~	–	–	✓	✓	✓	✓	~	✓
	ML-based models	✓	–	–	✓	✓	✓	✓	~	✓
	Stochastic models	~	–	–	✓	✓	✓	✓	~	✓

Legend: ✓ is clearly demonstrated in multiple reviewed implementations, or strongly demonstrated in at least one representative implementation; ~ is demonstrated only in limited cases, dependent on substantial adaptation, or primarily a theoretical capability; × generally absent or not demonstrated in the reviewed literature; and – is when the feature is not applicable to the model.

2.3 Machine learning for airspace modelling

With the rise in air traffic volumes and the introduction of new air vehicle types and operations in the airspace, the interest in effectively leveraging big aerospace data in airspace models is increasing. A common thread observed across much of the airspace modelling literature is that researchers are investigating the applicability of machine learning tools for improving modelling capabilities. This section introduces fundamental concepts on machine learning and its potential contributions to the analysis of data from airspace models.

2.3.1 Machine learning

Machine learning (ML) is a subfield of artificial intelligence (AI) that has emerged as a method of choice for developing practical software in many applications including computer vision, speech recognition, natural language processing (NLP), and robot control [320]. Mitchell [321] defines ML as a type of AI that uses a "*computer program that improves its performance at some task through experience*". ML computer systems can learn patterns from data and improve their performance at predicting a value over time without intervention from the designer. This ability for ML to learn from data has found widespread applications in finance for predictive analytics [322]; in healthcare for diagnostic accuracy [323]; and in retail for personalized shopping experiences [324]. Deep learning (DL) is a subset of ML comprised of a neural network, a type of machine learning with three or more layers [325]. DL models are particularly good at extracting high-level features from large volumes of data in applications such as speech recognition, image recognition, and NLP [326].

Figure 13 shows the four different types of ML: supervised ML, unsupervised ML, semi-supervised ML, and reinforcement learning. The difference between supervised and unsupervised ML is characterised by the use of labelled data. Supervised ML uses labelled data, where labelled data is defined as each input training variable being paired with a known target output label [327]. The goal of supervised ML is to learn a mapping from inputs to the target outputs [327]. In most applications, the trained model is used to predict the output associated with new inputs that were not part of the original input training set [327].

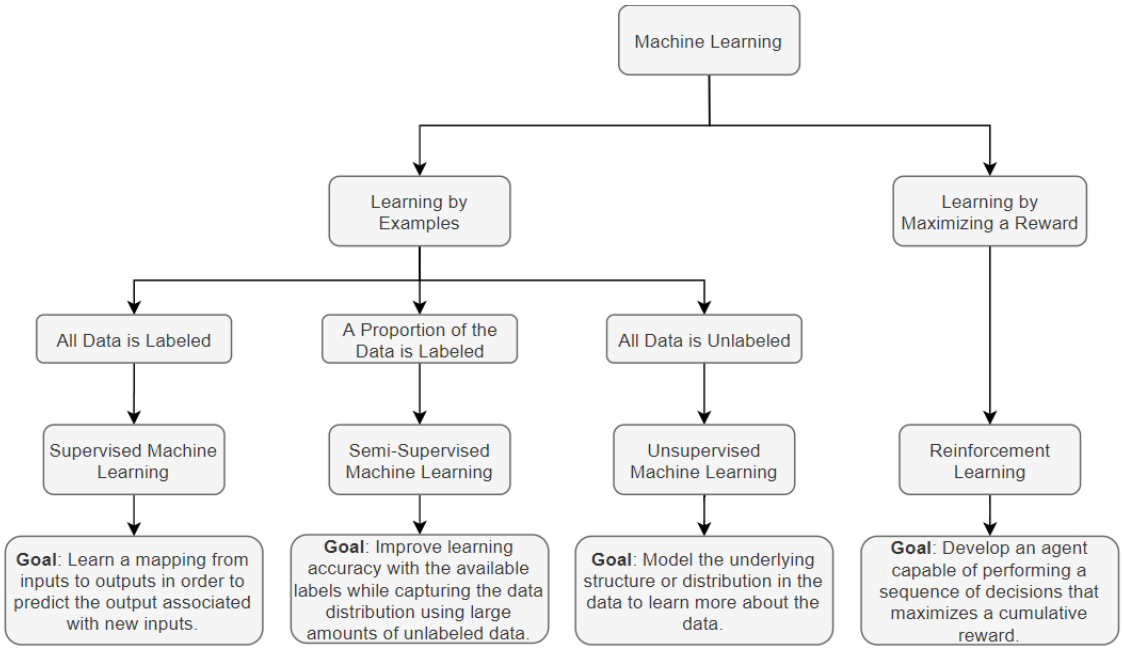


Figure 13: Types of machine learning

Unlike supervised ML, unsupervised ML uses data where all target outputs are unknown, and thus uses unlabelled data [328]. The objective of unsupervised ML is to train a model to capture the underlying structure or distribution of a dataset as a means of providing insights into the patterns contained within it [328].

Semi-supervised ML combines properties from both supervised and unsupervised learning, and uses both labelled and unlabelled data for training [329]. Typically, a small amount of labelled data with a large amount of unlabelled data is used [329]. This approach is useful when acquiring a fully labelled dataset is costly or labour-intensive. The goal of semi-supervised ML is to improve the learning accuracy based on the labelled data while at the same time capturing a more accurate data distribution using large amounts of unlabelled data.

Reinforcement learning (RL), rather than learning by examples from datasets like in supervised ML, unsupervised ML, and semi-supervised ML, learns by maximizing or minimizing a reward function [330]. Littman [331] describes RL as improving an agent's ability to make behavioural decisions based on experience. The agent interacts with the environment, receives evaluative feedback in the form of rewards or penalties, and adjusts its decision-making strategy to maximize long-term cumulative rewards, similar to learning by trial and error.

2.3.2 Machine learning algorithms

Figure 14 provides an illustration of selected algorithms commonly used in supervised and unsupervised ML. Semi-supervised ML can be implemented using algorithms from either supervised or unsupervised ML or a combination of both. Each algorithm is used to solve a defined problem as a means of achieving a specific goal. The following section will present a general overview of the literature on each algorithm for airspace modelling and Section 2.3.3 will discuss in more detail the literature specifically relevant to clustering algorithms.

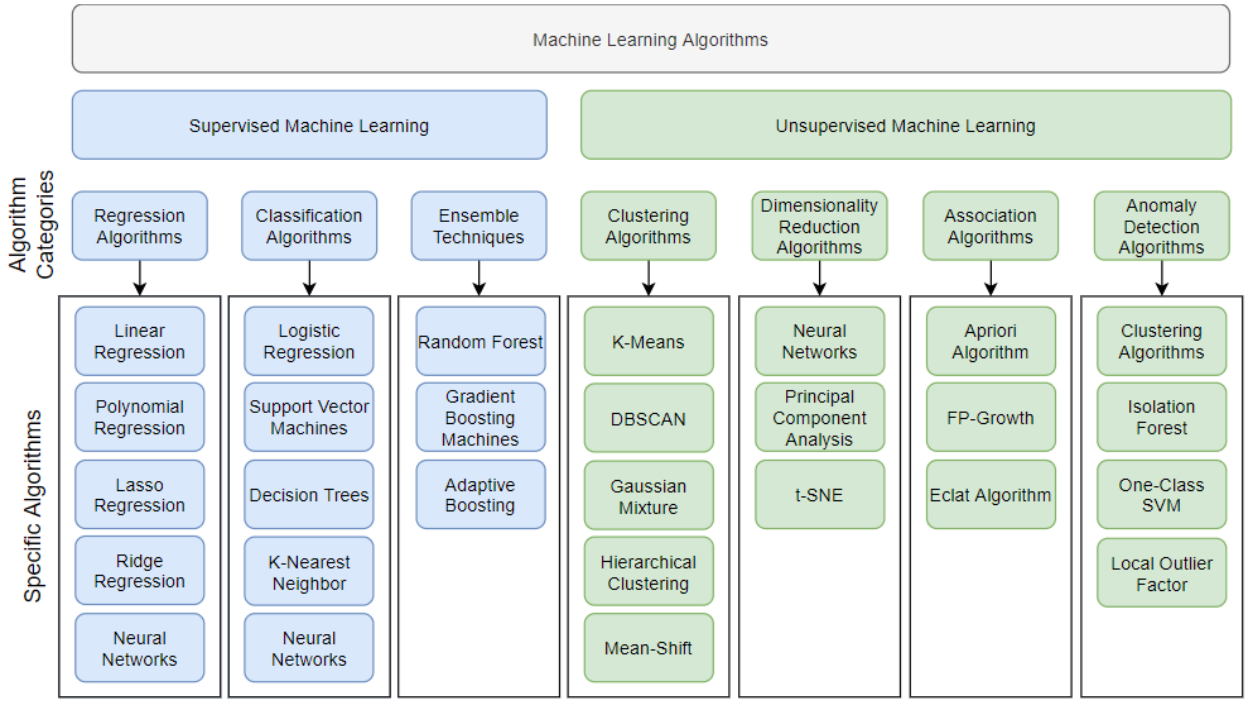


Figure 14: Supervised and unsupervised machine learning algorithms

Regression algorithms, such as linear regression and neural networks, are among the most commonly used type of machine learning and are used to predict continuous outputs by modelling relationships between dependent and independent variables, where the desired outputs are known ahead of time [332]. For example, regression techniques have been used to forecast flight fuel consumption, maintenance predictions, and turbulence levels in aviation. Chen et al. [333] applied nonlinear regression to model fuel consumption based on historical flight data. Raman et al. [334] developed a machine learning model for fuel flow and drag polar predictions using regression-type neural networks, used in improved aircraft trajectory planning. Al-Khanbashi and Jing [335] introduced an XGBoost regression model that utilized Quick Access Recorder (QAR) data and flight parameters to enhance real-time fuel flow predictions. In predictive aircraft maintenance, Hu et al. [336] used regression algorithms to develop a deep bidirectional recurrent neural networks (DBRNNs) ensemble method for predicting the Remaining Useful Life (RUL) of aircraft engines. The method extracts hidden features from sensory data and integrates multiple regression decision trees to improve prediction accuracy. Pebrianti et al. [337] applied a Convolutional Neural Network (CNN) to predict aircraft engine failures, using data from 21 aircraft sensors. They compared CNN with Recurrent Neural Networks (RNN) and demonstrated their superiority in predictive maintenance applications. For turbulence level prediction, regression algorithms were used by Sharma et al. [338] with RNNs, LSTMs, and GRUs to predict the Remaining Useful Life (RUL) of turbofan engines, providing insights into aircraft health monitoring and turbulence forecasting. Zhu et al. [339] combined CNN and LSTM models for life prediction of aircraft engines, allowing for the early detection of anomalies caused by turbulence and other external factors.

Classification algorithms focus on categorizing data into predefined classes (called categorical outputs) based on an input dataset [340]. These algorithms qualify as supervised ML since the desired categorical outputs are provided and paired with the input dataset (the output is known before training). Classification algorithms include logistic regression, support vector machines (SVMs), decision trees, k-nearest neighbors (k-NN), and classification-type neural networks. Classification-type neural networks have been used by Bosson and Nikoleris [341] to categorize aircraft trajectories for improved runway recognition. The authors used historical flight trajectory records, altitude, speed, and heading as input to predict output classes representing different runway approaches. The study achieved over 90% classification accuracy in predicting the correct trajectory category in under 40 seconds. Pérez-Castán et al. [342] used SVMs and decision trees in conflict detection models to predict potential separation infringements between aircraft. The input features include aircraft position, velocity, and altitude from ADS-B data and the classification output is a binary label: “Conflict” (risk of separation infringement) or “No Conflict” (safe separation maintained). This approach achieved 82% accuracy in correctly identifying conflicts. Truong and Choi [343] used the gradient boosting tree classification algorithm to predict the likelihood of airspace violations for a growing presence of small unmanned aircraft systems (sUAS). The model takes inputs like drone altitude, proximity to airports, and flight restrictions, categorizing violations into “Above 400 Feet”, “Within 5 Miles of an Airport”, “In Restricted Airspace”, or “Compliant”. This model achieved a 95.7% accuracy in predicting violations.

Ensemble techniques are supervised ML models that combines multiple base models to produce a more accurate and robust prediction than individual models [344]. Unlike traditional machine learning algorithms, which rely on a single model to learn patterns from data, ensemble techniques integrate multiple models to reduce variance and bias and improve generalization. The primary ensemble methods include Bagging, Boosting, and Stacking [345]. Bagging is an ensemble technique that trains multiple base models independently on different random subsets of the training data, using bootstrapping (sampling with replacement). The final prediction is made by averaging (for regression) or majority voting (for classification) over all models. Random Forest (RF) is a classic example of bagging that builds multiple

decision trees and combines their outputs to improve prediction stability and accuracy [346]. Boosting is an ensemble method where models are trained sequentially, with each new model correcting the errors of the previous ones. The final model is a weighted sum of all weak models, making it a strong predictor. Unlike bagging, boosting focuses on reducing bias rather than variance. Gradient Boosting Machines (GBM), Adaptive Boosting (AdaBoost), and Extreme Gradient Boosting (XGBoost) are examples of bagging ensemble techniques. Gradient Boosting Machines (GBM) iteratively train decision trees, optimizing a loss function to minimize errors [347]. Adaptive Boosting (AdaBoost) assigns higher weights to misclassified samples, ensuring subsequent models focus on difficult cases [348]. Extreme Gradient Boosting (XGBoost) improves GBM by adding regularization to prevent overfitting and enhance computational efficiency [349]. Stacking is an advanced ensemble method that combines predictions from multiple base models using a meta-model. Unlike bagging and boosting, which aggregate predictions directly, stacking introduces an additional model to learn how to best combine the predictions of different classifiers [350]. Zhou et al. [351] applied Gradient Boosting and Adaptive Boosting to predict airspace sectorization needs. The model used inputs such as historical traffic demand, capacity constraints, and congestion levels to classify sectors as *No Sectorization Required*, *Mild Sector Restructuring*, or *Full Sector Split*. Their approach improved air traffic control efficiency by providing real-time sector optimization. Kang and Jung [352] leveraged Random Forest and Gradient Boosting to classify terrain types for air-to-ground (A2G) communication optimization in UAV networks. Their model trained on digital surface model (DSM) geographic data, producing classifications such as *Urban*, *Rural*, *Mountainous*, and *Coastal*. Their findings showed that ensemble models improved communication reliability and classification accuracy. Zixuan et al. [353] trained a Random Forest algorithm using historical flight data to help dispatchers optimize fuel efficiency.

Dimensionality reduction algorithms are a type of unsupervised machine learning technique used to reduce high-dimensional datasets while preserving their key structures. The three primary methods for dimensionality reduction in machine learning are: Principal Component Analysis (PCA), t-Distributed Stochastic Neighbor Embedding (t-SNE), and neural networks in the form of Autoencoders. PCA is a linear transformation method that projects high-dimensional data onto a smaller set of principal components to capture the most variance in the dataset [354]. PCA is widely used in feature extraction, noise filtering, and visualization. t-SNE is a nonlinear dimensionality reduction technique primarily used for data visualization by mapping high-dimensional data into a 2D or 3D space while preserving local relationships between points [355]. Autoencoders are types of neural networks that compress and reconstruct data through encoding and decoding layers. Autoencoders are effective for handling nonlinear dependencies and noise reduction in high-dimensional datasets [356]. Navya and Bangalore [357] applied PCA to reduce redundant flight data variables in air quality measurements. The input features included meteorological parameters, flight path emissions, and congestion levels, which were transformed into a lower-dimensional space while preserving variance. Their study demonstrated that PCA-based clustering of airspace sectors improved traffic prediction and sectorization efficiency. Migenda et al. [358] developed a neural network-based PCA method to dynamically adjust the number of dimensions required for real-time conflict detection in air traffic control systems. Their approach continuously updated principal components based on live flight data, reducing computational costs while ensuring that crucial variance in aircraft trajectory data was retained. This adaptive model improved decision-making efficiency for air traffic controllers. Gupta et al. [359] utilized t-SNE in combination with autoencoders to identify anomalies in high-dimensional flight trajectory data. The model used aircraft speed, altitude, and weather conditions as inputs and classified flight patterns into normal and anomalous categories. Their study demonstrated that t-SNE enhanced trajectory visualization and made it easier to detect deviations from expected flight paths. Li and Wang [356] introduced Neural Principal Component Analysis (nPCA) for reducing redundant data in large-scale

geospatial airspace models. By combining PCA with deep learning autoencoders, their model achieved improved feature extraction from high-dimensional terrain data, aiding in the classification of airspace zones. Zhang et al. [354] proposed a PCA-based feature selection method to identify key traffic patterns in airspace sectorization models. Their algorithm removed redundant traffic attributes while retaining essential information, leading to improved classification of airspace congestion levels.

Association algorithms, sometimes referred as association rule mining, is a fundamental data mining technique used to uncover hidden relationships between items in large datasets. It works by identifying sets of items or features that frequently co-occur in the dataset. Three common association algorithms include: the Apriori algorithm, the Frequent Pattern Growth algorithm (FP-Growth), and the Equivalence Class Clustering and Bottom-Up Lattice Traversal algorithm (Eclat). The Apriori algorithm works by identifying frequent item sets through an iterative process of candidate generation and pruning. It is effective but computationally expensive due to its reliance on multiple dataset scans [360]. FP-Growth is a more advanced alternative to Apriori that eliminates the need for candidate generation by using a compact data structure called the FP-tree, improving computational efficiency [361]. Eclat [362] is a depth-first search algorithm that stores frequent item sets in vertical format, reducing memory usage and improving efficiency for large datasets. Rajpoot [362] implemented an enhanced Eclat algorithm to detect anomalous air traffic patterns in flight trajectory data. The algorithm processed input features such as aircraft speed, altitude, and deviation from expected flight paths to identify potential hazards. Compared to traditional Apriori-based methods, the improved Eclat algorithm significantly reduced computational time and enhanced detection accuracy. Lawal and Matthew [361] applied FP-Growth to optimize air traffic routes by analyzing historical flight data. The study extracted frequent route patterns based on weather conditions, congestion levels, and airspace restrictions. FP-Growth outperformed Apriori by reducing dataset scan times and improving processing efficiency by 82.04%. Jafarzadeh et al. [360] developed an Apriori-based fuzzy association rule mining approach to identify correlations between weather conditions and flight delays. The model used wind speed, precipitation, and visibility as inputs to categorize flights as *Delayed* or *On-Time*. By automating the minimum support selection, the algorithm effectively captured meaningful associations without discarding rare but important weather-related delay patterns. Ganjir and Chopra [363] proposed a hybrid Apriori-FP Growth model for dynamic airspace sectorization. The approach classified airspace regions into *High Traffic*, *Medium Traffic*, and *Low Traffic* sectors by analyzing frequent co-occurrences of flight density patterns. By integrating simulated annealing techniques, the model optimized air traffic flow, thereby improving efficiency in sector reallocation.

Anomaly detection algorithms are machine learning algorithms that are specifically designed to identify deviations from expected behaviour using an unsupervised learning approach. They learn patterns from normal data and then detect deviations from the normal data without requiring labelled anomalous data. Anomaly detection algorithms include: Isolation Forest (IF), One-Class Support Vector Machine (One-Class SVM), and Local Outlier Factor (LOF). IF is a tree-based algorithm that isolates anomalies by randomly partitioning the feature space, where anomalous data points are expected to be isolated more quickly than normal points [364]. The One-Class SVM is a model that learns a decision boundary around normal data points and classifies outliers based on this boundary [365]. LOF is a density-based method that compares the local density of a point with that of its neighbours to identify outliers [366]. Nguyen and Tran [364] applied Isolation Forest to detect anomalous flight trajectories in GNSS position time series data. The model analyzed latitude, longitude, and altitude variations to identify deviations from expected flight paths. The study demonstrated that Isolation Forest achieved a high ROC-AUC score, effectively distinguishing normal trajectories from anomalies. Russell et al. [365] evaluated One-Class SVM, Local Outlier Factor, and Isolation Forest for anomaly detection in IoT sensor networks used in airspace monitoring. Their results showed that One-Class SVM performed best at capturing rare but critical anomalies, while Isolation Forest

provided more robust generalization. LOF was effective in high-density areas but underperformed for sparse data distributions. Rawat et al. [367] implemented Isolation Forest for real-time anomaly detection in large-scale air traffic control sensor networks. The algorithm analyzed data streams from ADS-B sensors, identifying communication failures, GPS spoofing attempts, and radar errors. The model achieved 92% precision and reduced false positive rates compared to traditional statistical methods. Yang et al. [368] used One-Class SVM, Isolation Forest, and LOF to detect cyber-attacks targeting air traffic management systems. Their model analyzed real-time data integrity and identified fraudulent input signals that could compromise navigation or collision avoidance systems. The study concluded that Isolation Forest was the most robust method for detecting malicious intrusions. Shriram and Sivasankar [369] compared One-Class SVM, LOF, and Isolation Forest in anomaly detection for aerial surveillance data. The study used image-based data from satellites and drones to detect abnormal flight patterns and possible airspace intrusions. LOF excelled in detecting localized anomalies, whereas One-Class SVM provided better boundary definition for distinguishing normal and abnormal regions.

2.3.3 Clustering algorithms used in airspace modelling

Section 2.3.2 introduced clustering as one family of unsupervised machine learning algorithms. This section reviews the principal clustering approaches used in airspace-modelling studies and examines how their assumptions, computational characteristics, strengths, and limitations affect their suitability for different aerospace datasets.

Clustering algorithms are a type of unsupervised ML that can identify similarities between samples in a dataset as a means of identifying the underlying patterns in the data [370]. Clustering algorithms are trained to predict categorical outputs based on unlabelled data (the target outputs are unknown). Categorical outputs for clustering algorithms are called clusters (as opposed to the categories/class terminology used for classification algorithms) [371].

Among all the ML algorithms reviewed so far (regression algorithms, classification algorithms, ensemble techniques, dimensionality reduction algorithms, association algorithms, and anomaly detection algorithms*), unsupervised clustering algorithms* are particularly well-suited for analyzing big aerospace data found in data-driven airspace models because they:

1. **Do not require labelled data**: One of the primary advantages of clustering is that it is an unsupervised learning technique. Unlike classification algorithms, which require labelled datasets and predefined output classes, clustering works without prior knowledge of categories. This is particularly beneficial in air traffic analysis, where it is impractical to predefine all possible flight patterns, congestion zones, or airspace usage categories. Sáez-de-Cámara et al. [372] argue that unsupervised clustering provides practical advantages in anomaly detection by eliminating the need for costly data labelling, making it more scalable and adaptable to real-world air traffic data. Nicol and Puechmorel [373] applied unsupervised entropy-based clustering to aircraft trajectories, demonstrating that the method could autonomously group similar flight paths without human intervention.
2. **Are computationally efficient**: Clustering algorithms, especially partitioning-based and grid-based methods, offer computational efficiency advantages when compared to other ML algorithms. Corrado et al. [374] showed that clustering methods (specifically HDBSCAN) can process large-scale trajectory data more efficiently than classification models, making them ideal for real-time air traffic flow management. Unlike deep learning-based anomaly detection or classification models, which require extensive training and hyperparameter tuning, clustering methods can operate in real-time applications with minimal computational overhead [375]. Lawal and Matthew [361] compared FP-Growth association rule mining and clustering, finding that clustering methods outperformed association rule

mining in handling high-dimensional trajectory data due to their lower memory usage and faster computation times.

3. **Work across different dataset distributions and use cases**: Clustering algorithms work across different data distributions and structures, making them more adaptable than classification and regression models, which often assume specific data distributions; for example, regression algorithms work best for Gaussian distributions. This flexibility across different clustering paradigms makes clustering algorithms more robust for real-world air traffic data, which varies in shape, density, and complexity. The suitability of clustering depends on the dataset characteristics and the choice of clustering approach:
    - Partitioning-Based Clustering (KMeans) is effective for structured, spherical-shaped flight clusters and airspace sectorization [376].
    - Density-Based Clustering (DBSCAN, HDBSCAN) works well for irregular airspace patterns and detecting anomalous traffic congestion zones [377].
    - Hierarchical Clustering is useful for multi-level air traffic flow analysis, allowing dynamic grouping of flight patterns based on changing congestion levels [378].
    - Grid-Based Clustering is efficient for high-dimensional geospatial airspace analysis, reducing computational complexity [379].

4. **Have the ability to uncover hidden patterns and detect anomalies**: Unlike classification and regression models, which rely on predefined labels, clustering algorithms automatically discover hidden structures in data, making them ideal for identifying unknown or emerging airspace patterns and detecting anomalies [380]. Olive and Basora [381] combined clustering and anomaly detection to analyze historical aircraft trajectories, demonstrating that clustering can provide a taxonomy of normal and abnormal flight paths without requiring labelled training data. Chandra et al. [382] demonstrated that clustering improved airspace sub-sectorization by allowing adaptive adjustments based on real-time air traffic patterns, making it a more scalable and interpretable approach than supervised ML models. This ability to uncover hidden relationships in data is particularly valuable in dynamic airspace environments, where unknown congestion patterns, flight deviations, and emerging airspace usage trends must be detected without predefined assumptions.
5. **Interpretability**: Clustering models are recognized for their interpretability compared to other machine learning models, such as deep neural networks and complex ensemble models. Wang et al. [383] demonstrated that clustering-based preprocessing improved the interpretability of trajectory predictions in Terminal Manoeuvring Areas (TMA). By first clustering flight trajectories using DBSCAN and KMeans, the model created distinct traffic patterns that air traffic controllers could visualize and analyze more easily than using raw trajectory data or neural networks alone. In contrast, models like Multi-Cells Neural Networks (MCNN) and deep learning models demonstrated higher predictive accuracy but lacked intuitive visual outputs that controllers could directly interpret. Similarly, Zhou et al. [351] applied DBSCAN for air traffic flow prediction and dynamic airspace sectorization, leveraging the algorithm's clear spatial cluster outputs to visualize air traffic bottlenecks. The study emphasized that clustering models provided interpretable insights by allowing air traffic controllers to identify emerging congestion zones directly from visual data, whereas deep learning models required additional interpretability frameworks to explain their outputs. These examples highlight that while deep learning models often offer higher accuracy, clustering methods provide more interpretable and intuitive insights in airspace modelling applications.

Observation #24

Unsupervised machine learning clustering algorithms provide an objective and data-driven means to identify similarities between samples in a dataset and detect underlying patterns in the data using computationally efficient processes without requiring knowing the target inputs before training (it uses unlabeled data).

Observation #25

When compared to other machine learning algorithms like deep neural networks, examples from the literature have demonstrated that clustering algorithms produce results that are more easily interpretable.

Figure 15 shows the most used clustering algorithms. Clustering algorithms can be categorized based on their underlying approach to grouping data points and identifying patterns in datasets. Approaches used to cluster data include partitioning-based clustering; density-based clustering; model-based clustering (probabilistic clustering); hierarchical clustering; grid-based clustering; and fuzzy clustering (soft clustering).

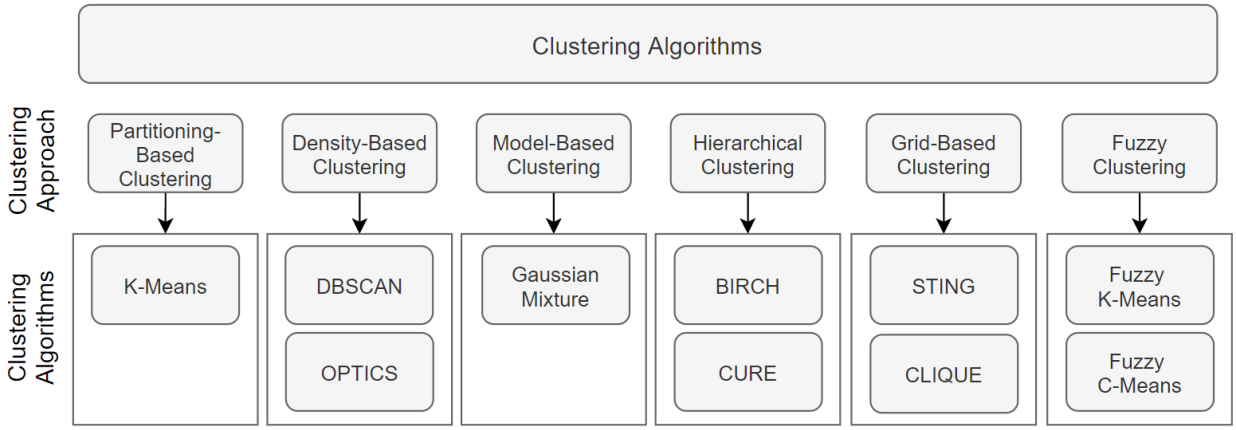


Figure 15: Unsupervised ML clustering algorithms

2.3.3.1 Partitioning-based clustering

Partitioning clustering algorithms divide a dataset into a fixed number of clusters by iteratively assigning points to clusters and refining the cluster boundaries. Each data point is assigned to a single cluster, and the goal is to minimize intra-cluster variance. They require users to specify the number of clusters beforehand.

KMeans is one of the most widely used unsupervised machine learning techniques for partitioning datasets into groups based on similarity [384], [385]. It partitions a dataset into clusters based on the mean distance between data points in the same cluster [386]. It does this by minimizing the variance within each cluster, often referred as the within-cluster sum of squares (WCSS). The algorithm starts by selecting 'k' points as the initial centroids, which are the initial "centers" of the clusters. Then, each data point in the dataset is assigned to the closest centroid based on a distance metric, typically the Euclidean distance. Once all data

points are assigned to clusters, the positions of the centroids are recalculated using the mean of all points assigned to each centroid's cluster. The assignment and update steps are repeated iteratively until the algorithm no longer improves significantly or a maximum number of iterations is reached. The simplicity, efficiency, and scalability of KMeans makes it a suitable choice for a wide range of clustering tasks where a rough grouping is sufficient. For very large datasets it may be required to use parallel computation frameworks. KMeans is sensitive to the initial choice of centroids at the first iteration [387]. This sensitivity is often addressed by running the algorithm multiple times with different initial centroids and choosing the best outcome [388]. An important aspect of using KMeans is deciding on the number of clusters 'k' to use, which is not determined by the algorithm itself and must be set before starting the algorithm. Methods like the elbow method [389] can help determine a suitable 'k' by evaluating the trade-off between the number of clusters and the WCSS.

KMeans has been widely applied in airspace modelling to categorize flight patterns and optimize air traffic flow. Zhong et al. [390] used a KMeans model to cluster trajectory characteristic points in terminal airspace, grouping turning points, altitude transitions, and speed adjustments to reveal operational patterns. Their method effectively identified congestion zones and helped improve airspace planning. Similarly, Bilen et al. [386] customized the KMeans algorithm to group aircraft in Aeronautical Ad-Hoc Networks (AANETs), accounting for factors such as latitude, altitude, and direction. This approach improved clustering stability and reduced cluster change rates by 75% compared to standard KMeans, enhancing connectivity success in high-density airspace. Govender and Sivakumar [391] employed KMeans for airspace pollution analysis, demonstrating its effectiveness in clustering environmental data to analyze pollution patterns in regions with dense air traffic. Jasra et al. [380] integrated KMeans with statistical models to develop a hybrid anomaly detection system for identifying irregularities in flight data, improving the detection of unexpected behaviour in complex airspace environments. Tang et al. [392] combined KMeans with the Time Warp Edit Distance (TWED) algorithm to group flight trajectories for 4D trajectory prediction. Their method improved prediction accuracy by clustering flight profiles based on matching patterns, supporting air traffic controllers in forecasting aircraft behaviour. These studies collectively highlight KMeans' versatility in identifying flight patterns, optimizing airspace zoning, supporting environmental analysis, and improving air traffic management efficiency.

2.3.3.2 Density-based clustering

Density-based clustering algorithms form clusters by identifying dense regions in the data, allowing for the detection of noise and irregularly shaped clusters. They do not require specifying the number of clusters and work well with non-spherical clusters.

Density-Based Spatial Clustering of Applications with Noise (DBSCAN) is a density-based clustering algorithm that identifies clusters based on the density of data points in a dataset, distinguishing between points that are in dense regions and those in sparse (noise) areas [393]. DBSCAN classifies points into three different types of points: core points, border points, and noise points. Core points have at least a minimum number of neighboring points within a given radius (ε), border points have fewer neighbors but are within a certain threshold from core points, and noise points are neither core nor border points [393]. The algorithm starts from a random point and checks if it is a core point. If it is, the cluster expands by recursively including all directly density-reachable points within the defined threshold [394]. The three main advantages of DBSCAN is that it doesn’t require specifying the number of clusters in advance; can find arbitrarily shaped clusters; and is robust against outliers [394]. A challenge faced when using DBSCAN is its parameter sensitivity. The quality of clustering depends on the settings of epsilon (radius of neighborhood) and minPts (minimum number of points in a neighborhood), which can be challenging to set without domain knowledge and can vary significantly with the density of the dataset [395]. The standard DBSCAN underperformed

for datasets where clusters vary widely in density [396]. Modifications and extensions to the standard DBSCAN were developed to address this issue [397]. DBSCAN is most used in exploratory data analysis where the data may contain outliers and noise, and the clusters are not uniformly sized or spaced [398]. Its ability to discover clusters of arbitrary shapes makes it well-suited for geographic, astronomical, and other spatial data applications [399]. It has also been used to segment images and identify shapes within them for image processing and in anomaly detection due to its ability to capture noise and outliers [400], [401].

DBSCAN has been widely employed in airspace modelling due to its ability to identify arbitrarily shaped clusters and detect noise points, making it ideal for analyzing air traffic patterns without requiring a predefined number of clusters. Gallego et al. [402] applied DBSCAN and its enhanced variation, Recursive DBSCAN (RDBSCAN), to cluster air traffic flows in dynamic airspace environments. Their study used flight trajectory data points (latitude, longitude, and altitude) as input features and generated clusters representing primary flight routes and peripheral air traffic paths. This method improved air traffic flow identification by distinguishing dense primary flight routes from sparse peripheral paths, making it easier to detect less commonly used air corridors. Alharbi et al. [147] used DBSCAN to analyze unmanned traffic management (UTM) airspace data, aiming to characterize UAV trajectories and identify deviations that indicate potential congestion points. Their model incorporated UAV flight path data, including position coordinates, time intervals, and mission status, as inputs. The outputs were clusters representing dense airspace regions and deviation points that flagged potential airspace congestion and safety risks. This clustering approach supported trajectory planning and enhanced safety analysis by isolating densely traveled UAV paths. Corrado et al. [374] employed HDBSCAN (a hierarchical extension of DBSCAN) combined with a weighted distance function to improve clustering performance for terminal airspace operations. Their method utilized ADS-B trajectory data as input, incorporating variables like speed, altitude, and sector boundary proximity. The algorithm outputted clusters that accurately grouped terminal airspace flows, successfully distinguishing converging and diverging air traffic patterns. By weighting data points closer to sector boundaries, their approach improved accuracy in detecting sector congestion and flow irregularities. DBSCAN's ability to isolate noise points has also proven valuable in anomaly detection. For instance, Deshmukh [403] applied DBSCAN for identifying abnormal flight behaviours in metroplex airspace operations. Using ADS-B flight surveillance data containing information on aircraft position, velocity, and altitude, the algorithm detected clusters representing normal traffic patterns and flagged noise points representing anomalous flight trajectories. This approach enabled early detection of aircraft deviating from standard flight paths, enhancing air traffic control safety protocols. These studies demonstrate that DBSCAN's flexibility, ability to detect noise, and suitability for arbitrarily shaped clusters make it effective for identifying airspace congestion, anomalous flight paths, and dynamic airspace patterns.

Mean Shift is a non-parametric clustering technique which does not require prior specification of the number of clusters and can handle complex cluster shapes [404]. Mean Shift iteratively shifts data points towards the densest area in a region based on kernel density estimation [405]. The algorithm automatically discovers the number of clusters based on the data distribution, with performance dependent on bandwidth selection [404]. Unlike many clustering algorithms that assume spherical clusters (like KMeans), mean shift can handle clusters of arbitrary shapes due to its flexibility in moving towards the highest density of data points [406]. Mean Shift is relatively robust to noise and outliers because it relies on the density of regions to locate and shift towards the most populated areas in the feature space [407]. On the other hand, it can be computationally expensive, especially with large datasets, as it requires multiple iterations to converge and each iteration involves recalculating distances to all other points. Its performance is sensitive to the choice of bandwidth parameter which determines the size of the region to search through [408]. Choosing an inappropriate bandwidth can lead to too many small clusters or a single large cluster. Like all other methods that are based on distance calculations, Mean Shift suffers from the curse of dimensionality, where its

performance degrades as the number of dimensions increases [409]. It is used in computer vision and image processing, particularly for tasks like image segmentation and object tracking [404].

The Mean Shift algorithm is used in airspace models due to its non-parametric nature, which allows for flexible clustering of flight trajectories based on data point density. This adaptability is particularly useful for flight data with the non-linear patterns often seen in historical data, where conventional algorithms may lack in performance. Peng et al. [410] employed Mean Shift to dynamically segment airspace sectors based on flight conflict data. Their model used aircraft position, velocity, and conflict risk metrics as input features. The algorithm's output successfully grouped flights into dense and sparse regions, allowing controllers to identify high-density traffic zones for improved workload balancing and sector partitioning. Levy [411] utilized Mean Shift to identify typical flight trajectories and characterize three-dimensional airspace patterns by clustering flight data using parameters such as latitude, longitude, and altitude. The algorithm's output revealed heavily trafficked corridors versus peripheral routes, which improved trajectory clustering and enhanced airspace planning. In radar data analysis, Zhu et al. [412] employed Mean Shift for radar signal pre-sorting to improve air traffic monitoring. Using radar signal characteristics such as signal strength, pulse frequency, and time of arrival as inputs, their model effectively grouped dense signal regions to isolate and improve radar signal detection. The output consisted of distinct clusters representing concentrated aircraft movements, reducing controller workload by improving data accuracy. Li and Hwang [413] demonstrated the Mean Shift algorithm's effectiveness for clustering real-time air traffic data to support dynamic airspace monitoring. Their model processed live data streams containing aircraft position, heading, and velocity. The algorithm's output generated adaptive clusters that identified regions with high air traffic density, improving decision-making support for air traffic controllers. Bepery et al. [408] proposed an Improved Mean Shift (IMS) algorithm, which incorporates a KD-tree structure and k-Nearest Neighbors (kNN) to enhance performance on dynamic datasets. Their method effectively handled missing values using Mean Imputation (MI) and improved clustering accuracy by 20% in large-scale datasets, making it suitable for dynamic airspace environments. These studies demonstrate that Mean Shift's ability to identify dense data regions without requiring a predefined number of clusters makes it effective in airspace sectorization, trajectory clustering, radar signal grouping, and real-time air traffic monitoring. Its non-parametric approach enables flexibility for managing complex, evolving airspace patterns.

2.3.3.3 Model-based clustering (probabilistic clustering)

Model-based clustering, sometimes also called probabilistic clustering, assumes that data is generated from a mixture of underlying probability distributions and attempts to estimate the parameters of these distributions. Data points belong to clusters with probabilities rather than hard categorical assignments (soft clustering).

Gaussian Mixture Models (GMMs) is a model-based clustering algorithm that models a dataset as a mixture of several Gaussian distributions [414]. This approach is especially relevant in unsupervised learning for clustering applications where the data is assumed to be generated from multiple Gaussian distributions, each representing a cluster. GMMs can accommodate clusters that have different sizes and covariances, unlike simpler methods like KMeans that assume clusters of similar shapes and sizes [415]. GMMs provide probabilistic cluster assignments, which can be more informative than hard assignments, offering insights into the data's underlying structure [414]. GMMs are not only used for clustering but also for density estimation, making them useful for generating new data points based on learned distributions. One challenge when using GMMs is that their results can be sensitive to the initialization of its hyperparameters, often requiring multiple runs to ensure robustness [327], [416]. GMMs can also be computationally expensive because of their use of the Expectation-Maximization (EM) algorithm for estimating the maximum likelihood parameters [417]. Determining the number of components (clusters) in a GMM is not

straightforward and often requires additional criteria like Bayesian Information Criterion (BIC) [418] or Integrated Completed Likelihood (ICL) [419] . GMMs are used in image processing by Xiong et al. [420] to segment images by modelling pixel distributions, taking into account the spatial correlations and varying intensities; in speech processing to model the distribution of audio features across different phonetic units [421]; and in the financial sector to model the behaviour of financial markets, capturing the multimodal nature of financial data distributions [422].

GMMs have been widely applied in airspace modelling due to their flexibility in clustering complex, overlapping flight paths and their ability to accommodate clusters of varying shapes and densities. Zeng et al. [423] employed GMM in combination with a deep autoencoder (DAE) to cluster aircraft trajectories in terminal airspace. Their model utilized high-dimensional trajectory data including latitude, longitude, altitude, and time stamps as inputs. The resulting clusters identified arrival and departure flow patterns, improving trajectory prediction and supporting airspace structure optimization. By reducing data dimensionality before applying GMM, their approach improved clustering accuracy in congested terminal airspace regions. Wu et al. [424] introduced the Sequential Possibilistic Gaussian Mixture Model (SPGMM), which incorporated time series data from continuous air traffic data streams. Their model utilized position, velocity, and time-ordered trajectory points as inputs to predict pattern changes in flight behaviour. By detecting early deviations from typical flight patterns, SPGMM improved the detection of irregular or potentially hazardous airspace behaviours important in air traffic control. Paek et al. [425] used GMM to predict en-route arrival times for aircraft traveling through congested airspace. The model took historical trajectory data, including arrival times at significant waypoints, as input. The output was a probability distribution of expected arrival times, enhancing prediction accuracy in busy flight corridors and allowing air traffic controllers to better assess uncertainty in flight delays. Min et al. [426] employed GMM in combination with knowledge transfer techniques to enhance airspace monitoring and aircraft design analysis. Their model utilized flight performance metrics, weather data, and aircraft design parameters as inputs to create probabilistic clusters that identified critical performance patterns under various flight conditions. By modelling the relationships between these variables, their method improved predictive maintenance strategies and enhanced flight path deviation predictions, contributing to improved risk assessment in complex airspace environments. In the field of UAV (Unmanned Aerial Vehicle) navigation, Yao et al. [427] applied GMM with receding horizon control (RHC) to improve UAV search and rescue operations in complex environments. Their model processed target probability distribution maps as input and generated clusters representing high-probability search regions. This method improved UAV search efficiency by focusing efforts on the most promising target locations. Zhao et al. [428] employed GMM for incremental anomaly detection in flight data recorder (FDR) datasets. Using aircraft position, engine performance, and control inputs as features, their model produced clusters representing standard flight behaviours while isolating deviation points as potential anomalies. This approach enabled real-time anomaly detection in dynamic air traffic systems by continuously adapting the GMM model using newly arriving data. These studies collectively demonstrate that GMM's probabilistic clustering approach is effective in trajectory clustering, arrival time prediction, anomaly detection, and UAV navigation.

2.3.3.4 Hierarchical clustering

Hierarchical clustering uses a tree-like structure to group data points based on their similarities [429]. There are two main types of hierarchical clustering: Agglomerative (bottom-up) and Divisive (top-down). The Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH) algorithm is an example of agglomerative clustering. BIRCH incrementally constructs a tree called the Clustering Feature Tree (CFT) while condensing the dataset. The tree represents data points in a way that attempts to minimize the information loss and can be rebuilt dynamically as new data comes in, making it efficient for dynamic

clustering environments [430]. BIRCH is noted for its efficiency in handling very large datasets due to its ability to incrementally and dynamically cluster incoming multi-dimensional metric data [431]. This makes BIRCH suitable for applications like real-time data clustering and situations where memory is a constraint [431]. The performance of BIRCH is sensitive to its two primary parameters: the threshold and the branching factor, where incorrect parameter settings can lead to suboptimal clustering results [432]. While BIRCH is effective for spherical-shaped clusters, its performance may degrade with non-spherical shapes or varying densities within the data [433]. Its effectiveness decreases as the dimensionality of the dataset increases, which is a common limitation in many clustering algorithms [434]. BIRCH is used in customer segmentation to identify groups of customers with similar behaviours for targeted marketing [435]; in genomic data analysis to group genes with similar expression patterns to identify functional communities [436]; and in document clustering for organizing a large corpus of text documents into hierarchies of topics [437].

Hierarchical clustering has been applied in airspace modelling, particularly for organizing flight data into structured trees that reflect hierarchical air traffic flows. Deng et al. [438] applied agglomerative hierarchical clustering combined with dynamic time warping (DTW) to identify deeply embedded trajectory patterns in vectored airspace. Using ADS-B surveillance data from Incheon International Airport (South Korea), their model took latitude, longitude, altitude, and timestamp as inputs to generate clusters that accurately reflected key arrival patterns to identify high-density flight corridors. Corrado et al. [374] combined hierarchical clustering with a weighted Euclidean distance function to improve trajectory clustering in terminal airspace. Their method utilized ADS-B trajectory data, incorporating parameters such as position, speed, and altitude. The output consisted of hierarchical clusters that segmented air traffic flows into major arrival/departure corridors and less frequent peripheral paths. Olive and Basora [381] employed hierarchical clustering to detect anomalous flight behaviours in past en-route Mode S trajectories. Using ADS-B flight data containing position, speed, and heading as inputs, their model constructed hierarchical clusters to isolate standard air traffic patterns while detecting deviation points as potential anomalies. This method provided insights for identifying deconfliction maneuvers, weather-related diversions, and potential collision risks. In another example, Chu et al. [439] developed a clustering ensemble method using hierarchical clustering integrated with a similarity matrix. The model employed flight trajectory data from Nanjing Lukou Airport (China), combining altitude, heading, and waypoint sequences as inputs. The resulting hierarchical clusters identified frequent flight patterns and optimized airspace sectorization. Zhao and Sartipi [440] applied a hierarchical clustering model for identifying anomalous flight behaviours based on trajectory similarity metrics. Their model took taxi GPS trajectory data as inputs to identify anomalous routes deviating from typical driving behaviours. The system successfully detected traffic anomalies, demonstrating hierarchical clustering's value in identifying unusual patterns across different types of transportation data. These studies collectively demonstrate that hierarchical clustering — including agglomerative methods and extensions like BIRCH — is effective for trajectory clustering, congestion detection, and anomaly identification. By grouping data in hierarchical structures, these models provide scalable solutions for dynamic air traffic environments, supporting both high-level traffic flow management and localized operational adjustments.

2.3.3.5 Grid-based clustering

Grid-based clustering methods divide the data space into a finite number of grid cells and then groups spatially dense cells into clusters. Grid-based clustering is more computationally efficient than partitioning or density-based methods because it aggregates data into grids (rather than using individual data points) and scales efficiently with large datasets. Two notable grid-based clustering algorithms are STING (Statistical Information Grid) and CLIQUE (Clustering In QUEst).

STING divides the data space into a hierarchical grid structure. Each grid cell stores statistical parameters (mean, variance) about the data points within it. STING processes data by merging, splitting, or refining grid cells based on these parameters to efficiently detect clusters. STING's hierarchical structure allows for fast query response times and scalability to large datasets, making it ideal for real-time airspace data analysis. Saini and Rani [441] provided an extensive overview of STING's application in air traffic analysis. They emphasized STING's efficiency in segmenting large-scale flight trajectory datasets. By clustering flight trajectories using latitude, longitude, and altitude as inputs, their approach identified high-density traffic corridors and potential congestion zones. The output clusters supported dynamic airspace management by improving the identification of air traffic flow patterns.

CLIQUE combines grid-based clustering with density-based principles, effectively identifying clusters in high-dimensional data. It partitions the data space into non-overlapping grid cells and detects dense regions as clusters. CLIQUE is particularly useful for analyzing data with multiple attributes like aircraft speed, altitude, and heading. CLIQUE efficiently handles datasets with numerous attributes by reducing dimensionality and detecting meaningful patterns in complex datasets. He et al. [442] improved the classic CLIQUE algorithm by introducing automatic grid splitting and integrating it with KMeans clustering. Their model used flight trajectory data, including position, altitude, and timestamp, to cluster air traffic patterns. Their improved CLIQUE algorithm successfully identified flight path clusters with 36.9% higher accuracy than the original CLIQUE method, particularly in identifying complex flight behaviour patterns and separating overlapping traffic flows.

Other examples of grid-based clustering include CONGEST, introduced by Parmar et al. [443] as a novel grid-based clustering algorithm designed to predict congestion zones for UAVs in dynamic airspace environments. Using UAV flight plan data (including waypoints, altitude, and timestamp), CONGEST identified high-traffic congestion zones. The algorithm demonstrated high accuracy (91.5% - 98.5%) in detecting congested areas within seconds, providing a fast and effective solution for UAV traffic management. Ayhan and Samet [51] applied a grid-based airspace partitioning model to enhance aircraft trajectory prediction. Their model divided the airspace into 3D grid cells that incorporated weather data, wind conditions, and flight position points. This technique enabled real-time trajectory prediction by aligning aircraft paths to weather-influenced airspace zones. The resulting clusters improved accuracy in identifying potential delays and congestion points. Cai et al. [444] designed a grid-based clustering method using the *Multi A algorithm** to manage dynamic airspace during large-scale battlefield air operations. Their model used 3D airspace grids to identify high-density mission zones and congested flight routes. The output provided air traffic controllers with optimized flight corridors, improving operational efficiency during complex combat missions.

Observation #26

Clustering algorithms demonstrate varying degrees of scalability in handling large-scale airspace datasets. While grid-based and partitioning-based methods like K-Means are known for their computational efficiency, density-based methods like DBSCAN and hierarchical clustering face performance bottlenecks in large-scale applications and may require additional modifications to remain efficient.

2.3.3.6 Fuzzy clustering

Fuzzy clustering methods assign data points to multiple clusters with varying degrees of membership (soft clustering), making them particularly useful for airspace modelling, where uncertainty and overlapping data

patterns are common. Unlike traditional hard clustering methods that enforce distinct cluster boundaries, fuzzy clustering methods provide flexible solutions for dynamic and ambiguous datasets. Fuzzy KMeans and Fuzzy C-Means (FCM) are two examples of fuzzy clustering methods.

Fuzzy KMeans is an extension of the traditional KMeans algorithm that assigns each data point a membership score in multiple clusters rather than strictly assigning each point to one cluster. This soft assignment is calculated based on the inverse distance to each cluster centroid, with closer points receiving higher membership values. This makes Fuzzy KMeans more effective when compared to traditional KMeans for handling overlapping clusters, making it suitable for airspace models with ambiguous flight patterns or uncertain trajectory data. Choong et al. [445] employed both Fuzzy KMeans and Fuzzy C-Means to cluster vehicle trajectories for traffic pattern analysis. While the study focused on land-based transportation, the methodology was transferable to airspace modelling. Using trajectory data, time-series data, and position information as inputs, FKM and FCM was used to group routes with ambiguous boundaries, offering insights into congestion points and common movement patterns.

Similar to Fuzzy KMeans, FCM assigns points to multiple clusters with membership scores. The algorithm minimizes a weighted objective function that balances membership values and cluster compactness. Unlike Fuzzy KMeans, FCM excels in detecting non-linear clusters and data points in complex spatial distributions, improving precision in high-density and uncertain environments. Chen [446] applied FCM combined with Kalman Filtering (KF) for tracking moving aircraft in congested flight zones. Their model used ADS-B flight data (including position, altitude, and velocity) as inputs to predict movement patterns and identify regions of high-density flight activity. The system detected abnormal runway intrusion events in busy airport environments. Akinin et al. [447] implemented FCM for clustering UAV aerial photography data to create efficient flight paths for remote sensing and mapping tasks. Their model processed GPS coordinates, altitude, and image data as inputs. By segmenting UAV data into multiple regions with overlapping boundaries, the algorithm improved path efficiency and minimized redundancy during data collection. Cao and Wang [448] introduced a Density-Based Fuzzy C-Means Multi-Center Re-Clustering (DFCMRC) algorithm for radar signal detection in airspace monitoring. Their model combined FCM with DBSCAN principles to enhance radar signal grouping in dynamic airspace conditions. Using pulse data, signal amplitude, and frequency data as inputs, the algorithm improved radar signal sorting accuracy by 30% compared to traditional methods.

2.3.3.7 Summary of the reviewed clustering algorithms

Table 10 summarizes the reviewed clustering algorithms including example algorithms; their ideal dataset distribution applications; and their strengths and weaknesses. Two main observations can be made about the choice of clustering algorithm for a pre-determined application.

Observation #27

A dataset's distribution is a driving factor in deciding which clustering algorithm is most effective for the intended application.

Observation #28

Each clustering algorithm has strengths and weaknesses that can be used for choosing which algorithm to use depending on the intended application.

Table 10: Summary of the clustering algorithms

Category	Ideal distribution	Examples	Strengths	Weaknesses
Partitioning-Based	• Spherical (or globular) clusters with relatively equal sizes and uniform density. • Minimal overlap	• KMeans • K-Medoids	• Fast and scalable, making it ideal for large datasets • Efficient for datasets where clusters are spherical and evenly sized. • Simple implementation with strong interpretability in visualizing dataset patterns.	• Needs predefined k • Assumes clusters are spherical, making it less effective for irregularly shaped data distributions. • Highly sensitive to initial centroid placement • Sensitive to outliers
Density-Based	• Irregular, non-spherical patterns. • Arbitrarily shaped clusters (curved, elongated, or non-convex clusters) with variable densities.	• DBSCAN • HDBSCAN • OPTICS • Mean shift	• Excels at detecting clusters of irregular shapes (curved or elongated flight paths). • Does not require users to predefine the number of clusters. • Effectively identifies noise and unusual flight patterns.	• Struggles with varying densities • Its performance is highly sensitive to parameter tuning (ϵ and MinPts).
Model-Based	• Similar to multiple Gaussian (normal) distributions • Elliptical or Overlapping Clusters with varying sizes, densities, or complex relationships.	• GMM • HMM	• Highly flexible, capable of modelling clusters with elliptical or irregular shapes. • Provides probabilistic membership values, enabling uncertainty modelling.	• Computationally intensive • Prone to convergence in local optima, requiring careful initialization.
Hierarchical	• Nested, multi-level data structures • Clusters with varying sizes	• Agglomerative • Divisive	• No need to predefine the number of clusters. • Captures multi-level data relationships (hierarchy)	• High memory and computational cost • Difficult to modify the clustering structure once the tree is built.
Grid-Based	• Uniformly distributed data in a structured space with defined boundaries.	• STING • CLIQUE	• Scalable for large datasets • Computational efficiency • Robust for spatial data • Fast with uniform data distribution	• Requires careful grid resolution choice • Parameter sensitivity • Inefficient for irregular cluster shapes • Limited adaptability to dynamic data
Fuzzy Clustering	• Overlapping or ambiguous clusters where points may belong to multiple categories.	• Fuzzy C-Means	• Allows data points to belong to multiple clusters with different degrees of membership • Handles uncertainty and overlap • Effective for Uncertain or Noisy Data	• Large computational complexity • Parameter sensitivity • Potential for ambiguity

2.4 Synthesis of the literature review observations and research gaps

The preceding literature review identified 28 detailed observations concerning aerospace big data, airspace modelling, and machine learning. These observations provide traceability to the specific findings discussed throughout Chapters 1 and 2, but several address closely related aspects of the same broader modelling challenges. To clarify their relationship to the research undertaken in this thesis, the detailed observations are synthesized below into five summary observations from which the research gaps are derived.

Summary observation 1: Aerospace data availability has increased, but its direct usability remains limited

The literature documents substantial growth in the volume, velocity, and variety of aerospace data, including the increasing availability of open surveillance and environmental datasets. However, these datasets cannot generally be used directly for airspace modelling. They require collection, cleaning, filtering, transformation, temporal alignment, spatial referencing, and other processing before they can support analysis. Their varying scale and heterogeneity also create computational challenges, particularly when long historical periods or multiple data sources are considered. This summary observation is principally informed by Observations #1–3, #7–8, and #11–12.

Summary observation 2: Modern airspace is three-dimensional, time-varying, heterogeneous, and application-dependent

The reviewed airspace models represent a complex environment containing aircraft, weather, airspace structures, infrastructure, and other spatially referenced entities. These entities can differ in their physical characteristics, operational roles, and rates of change. The literature also includes both conventional and unconventional aircraft, although models jointly representing heterogeneous aircraft operations remain less common than models developed for a limited set of vehicle types. Simplified two-dimensional or static representations remain useful for some applications, but comparative studies show that explicitly representing altitude and temporal variation is important when the analytical objective depends on three-dimensional position or changing operational conditions. This summary observation is principally informed by Observations #4–6 and #9–17.

Summary observation 3: Location-based and DGGS representations can provide a common structure for heterogeneous airspace data, but reusable implementations remain limited

The reviewed literature is dominated by trajectory-based models, which associate outputs with individual aircraft paths, and region-based models, which aggregate airspace conditions within predefined sectors or zones. Location-based approaches also exist and provide a means of assigning observations and derived values to fixed spatial locations over time. DGGS frameworks offer one structured implementation of this paradigm through hierarchical, indexed spatial cells. Figure 16 shows visual depictions of the three spatial modelling paradigms and their outputs.

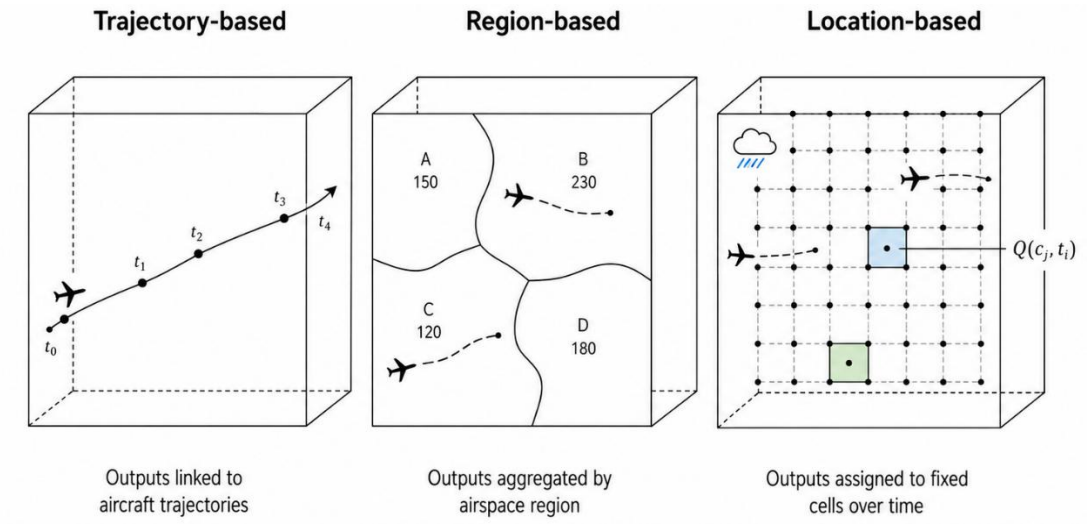


Figure 16: Depictions of the three modelling paradigms

DGGS and related grid systems have been applied to selected aerospace problems, including data indexing, trajectory representation, conflict detection, path planning, and airspace sectorization. However, their use remains comparatively limited within the reviewed airspace modelling literature. The identified implementations are generally developed around one principal data type or one analytical objective. Evidence is particularly limited for models that use a common DGGS-based 4D structure to integrate heterogeneous aircraft and environmental observations and then reuse that structure across multiple downstream analyses. This summary observation is principally informed by Observations #6 and #22.

Summary observation 4: Airspace phenomena occur across multiple spatial and temporal scales

Terminal-area and system-wide airspace models operate at substantially different geographical resolutions and levels of detail. Terminal models generally require fine spatial and temporal resolution to represent individual aircraft operations, whereas system-wide models commonly use coarser representations to support traffic-flow and capacity analysis over larger regions. Aircraft and environmental phenomena similarly evolve at different temporal resolutions. Multi-resolution approaches have been proposed to reconcile these requirements, but models that integrate multiple spatial and temporal scales within a common framework remain less prevalent. This summary observation is principally informed by Observations #14–20.

Summary observation 5: The analytical value of airspace models depends on the derived products and interpretation methods applied to their contents

Airspace models do not provide analytical insight solely by organizing, indexing, or visualizing data. Their utility depends on whether application-relevant and operationally meaningful quantities can be derived from their contents and interpreted consistently across space and time. The reviewed literature contains many models that produce outputs for individual applications but provides more limited evidence of reusable airspace representations supporting multiple derived analytical products.

Historical airspace data also contain recurring spatial, temporal, and operational structure that can provide information beyond the direct organization or visualization of individual observations. However, identifying and interpreting this structure directly from the data is challenging when the relevant behavioural groupings are not known in advance. The broader airspace-modelling literature further shows

that model outputs can be influenced by predefined assumptions, constraints, and subject-matter expertise, while supervised analytical approaches require predefined target classes. Consequently, methods that derive structure directly from observed data can reduce reliance on a priori behavioural definitions while retaining domain knowledge for subsequent interpretation and validation. This summary observation is principally informed by Observations #21–25 and #27–28.

The summary observations lead to four research gaps.

Research gap 1: Reusable multi-source 4D airspace modelling

Research Gap #1

Existing airspace models are often specialized by aircraft or data type, geographical scale, dimensional representation, or analytical application, limiting their modular reuse across different airspace analyses.

Existing airspace-modelling approaches are commonly specialized according to particular aircraft or data types, spatial representations, geographical regions or scales, dimensional representations (two, three, or four dimensional), and intended analytical applications. For example, some models are developed for specific vehicle classes, fixed terminal or en-route regions, two-dimensional rather than three-dimensional analysis, or a single predefined analytical task. Although these specialized models can perform their intended functions effectively, their underlying representations are often closely coupled to those assumptions and applications. This limits their modular reuse when the vehicle type, data source, spatial scale, dimensional requirements, or downstream analytical objective changes.

The resulting gap is the need for a more modular and reusable four-dimensional airspace representation that can accommodate heterogeneous airspace entities within a common three-dimensional and time-varying structure, adapt across different spatial scales and regions, and preserve the information required by distinct downstream analyses.

DGGS and related hierarchical grid approaches provide one potential means of addressing this modelling problem because they support indexed, location-based representation, multiple spatial resolutions, and consistent association of heterogeneous georeferenced data. Existing aviation applications demonstrate these capabilities individually, but the reviewed implementations generally remain focused on particular data types or analytical functions. These characteristics suggest that a DGGS-based location representation could provide a persistent common airspace structure that is less tightly coupled to any one analytical application. This observation motivates the evaluation of DGGS as the proposed spatial framework in Hypothesis #1.

Research gap 2: Computational efficiency of location-based, data-driven 4D airspace models

Research Gap #2

Limited comparative evidence exists on the computational efficiency of location-based, data-driven 4D airspace models when information required for downstream analyses must be preserved.

The literature identifies algorithmic optimization, feature selection, dataset partitioning, parallelization, spatial indexing, and improved data-management strategies as potential means of increasing computational efficiency. However, the computational behaviour of location-based, data-driven 4D airspace models is less well characterized under controlled conditions in which the analytical requirements of the model are held constant. In particular, no comparisons evaluate whether reductions in runtime or memory demand are achieved while preserving the model information required for subsequent analyses for location-based models. Without this distinction, an apparent efficiency improvement may result from reducing or altering the analytical task rather than processing the required information more efficiently.

Research gap 3: Reuse of airspace models for multiple analytical products

Research Gap #3

Limited evidence exists on how a reusable location-based 4D airspace model can be used to compute and interpret derived analytical products at common three-dimensional locations and time steps, particularly when those products depend on multiple heterogeneous, time-varying airspace entities.

The reviewed airspace models generate a range of analytical outputs, including predicted trajectories, congestion estimates, conflict alerts, capacity forecasts, weather-avoidance fields, and spatial risk indicators. However, these models are commonly developed around one predefined analytical objective, closely coupling the underlying airspace representation to the output it is intended to produce. Location-based studies also demonstrate that individual observations or derived quantities can be associated with spatial points or cells, but reuse of the same heterogeneous 4D representation to produce different analytical products is less established. The unresolved issue is therefore whether a common multi-source airspace representation can serve as a reusable analytical foundation from which multiple derived quantities can subsequently be computed and interpreted at common three-dimensional locations and time steps. This concerns analytical reuse of the model after construction rather than initial organization of its source observations (Gap #1).

Research gap 4: Data-driven identification and interpretation of operational patterns in airspace model data

Research Gap #4

Methods for identifying and interpreting recurring operational patterns directly from airspace model data without requiring the relevant groupings to be predefined through subject matter expert assumptions or application-specific constraints, are not well established.

Historical airspace model data can contain recurring spatial, temporal, and operational structure that is not immediately apparent from individual observations. However, identifying and interpreting that structure is challenging when the relevant groupings are not known in advance. Existing analytical approaches can require categories, rules, labels, or analytical boundaries to be defined using subject-matter expertise or application-specific assumptions before the underlying structure of the data has been examined.

The resulting gap concerns methods capable of identifying meaningful recurring structure directly from airspace model data and subsequently supporting its interpretation without requiring the operational groupings themselves to be defined a priori. Domain knowledge remains important for interpretation and validation, but the initial group structure should be derived from the observed data rather than imposed before analysis.

These four gaps provide the basis for the research questions, hypotheses, and validation strategy presented in Chapter 3.

Chapter 3: Research approach

This chapter presents the research questions, hypotheses, and validation strategy developed in response to the four research gaps presented in Section 2.4.

The research approach is organized into three experiments as summarized in Figure 17. Experiment #1 addresses Research Gaps 1 and 2 by developing a modular and reusable 4D airspace model and evaluating its computational efficiency while ensuring the preservation of the model information required for downstream analyses. Experiment #2 addresses Research Gap 3 by evaluating whether the reusable model can support the derivation and interpretation of multiple analytical products, using location-based multi-hazard risk analysis as the selected case study. Experiment #3 addresses Research Gap #4 by evaluating whether recurring operational patterns can be identified from trajectory-level airspace model outputs in a data-driven manner without defining the relevant groupings in advance. Identified patterns are interpreted using post-clustering domain evidence in the context of a behaviour-based aircraft categorization case study.

Together, the three experiments operationalize the main research objective of the thesis stated in Section 1.5: to develop a methodology for a data-driven airspace modelling approach that leverages big aerospace data using collection and processing techniques effective at managing large-scale datasets while optimizing computational resources. Experiment #1 establishes the reusable DGGS-based 4D airspace model and evaluates its computational practicality. Experiment #2 evaluates whether heterogeneous data organized within that model can subsequently be transformed into interpretable location-based analytical products. Experiment #3 evaluates a second and fundamentally different analytical use by applying unsupervised machine learning to trajectory-level airspace-model outputs. Together, the experiments therefore evaluate the proposed methodology from model construction and computational execution through two distinct downstream analytical applications.

The remaining sections of this chapter describe these three research questions, hypotheses, and experiments.

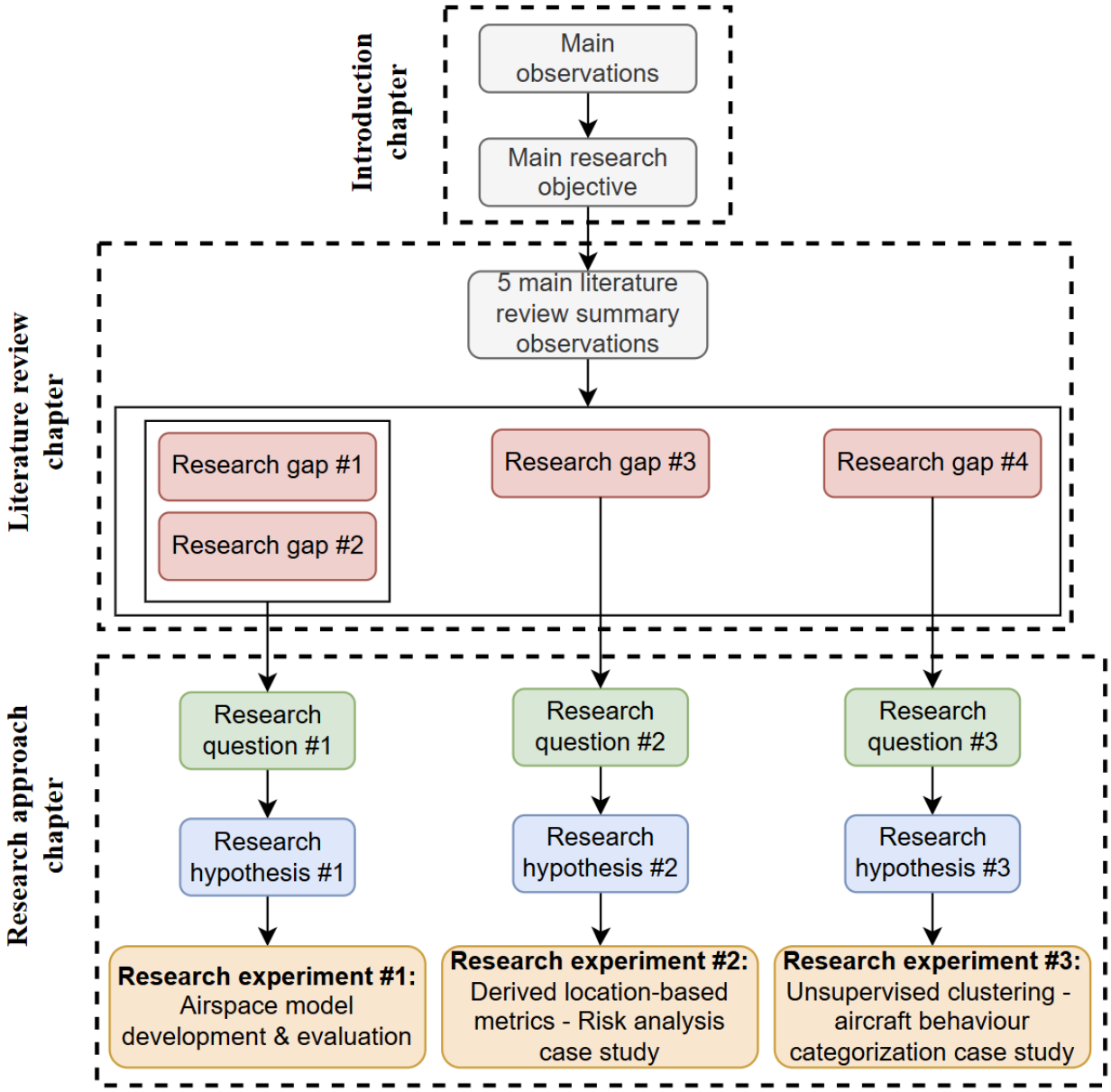


Figure 17: Research approach

3.1 DGGS-based airspace model development and computational evaluation

The first two research gaps motivate the first research question and corresponding hypothesis and experiment. Research Gap #1 concerns the specialization of existing airspace models by aircraft or data type, geographical scale, dimensional representation, or analytical application, which limits their modular reuse across different analyses. Research Gap #2 concerns limited studies on the computational efficiency of location-based, data-driven 4D airspace models when the information required for downstream analyses must be preserved.

These gaps require both a modelling and a computational response. The modelling response must provide a common spatial and temporal representation capable of integrating heterogeneous historical airspace data while remaining reusable across different applications. The computational response must evaluate whether that representation can be generated and reused efficiently without changing or removing information required by downstream applications. This leads to the first research question:

Research question #1

How can historical aerospace data be integrated into a generalized, reusable, data-driven 4D airspace model and processed in a computationally efficient manner while preserving the information required for downstream model reuse and analysis?

This thesis hypothesizes that a Discrete Global Grid System (DGGS) is an appropriate spatial framework for the proposed airspace model because it provides a structured way to divide space into discrete cells. As discussed in the literature review, this framework is useful for geospatial modelling because it supports spatial indexing, multi-resolution representation, and the consistent mapping of different georeferenced datasets into a common structure. In the context of this thesis, the DGGS framework is used to represent the airspace as a set of three-dimensional cells that can also be indexed over time.

The literature review also identified computational efficiency as a key design requirement for data-driven airspace models. Strategies to improve computational efficiency include algorithmic optimization, parallel processing, database optimization, spatial indexing, feature selection, and dataset partitioning. This thesis hypothesizes that feature selection and dataset partitioning could effectively address two practical causes of computational cost in the proposed workflow. Feature selection reduces the number of variables carried through the processing pipeline by retaining only those required for a specific modelling task. Dataset partitioning divides large datasets into smaller subsets that can be processed more efficiently, then recombined to produce the results.

Hypothesis #1

A Discrete Global Grid System (DGGS) framework can integrate historical aircraft, weather, and other time-varying 3D airspace data into a single reusable 4D airspace model. Feature selection and dataset partitioning will improve the computational practicality of the modelling workflow by reducing the amount of data processed or held in memory while preserving the model outputs required by downstream analyses.

Research Experiment #1 tests Hypothesis #1 in two parts. In the first part, historical ADS-B aircraft trajectories and weather radar data are collected, processed, transformed into a common geospatial

reference system, and mapped into a 4D DGGS airspace model. The resulting outputs include processed and georeferenced airspace datasets, DGGS-indexed aircraft and weather data, and visualizations that verify the model's spatial and temporal structure.

The second part of Research Experiment #1 tests the computational efficiency of the model development workflow. The model is evaluated both with and without feature selection and dataset partitioning, across different airspace configurations and varying dataset sizes. The resulting outputs include metrics such as total wall-clock time, stage-level runtime, memory usage, throughput, and relative speedup. The experiment is considered successful if the methodology can integrate the selected public aerospace datasets into a common 3D, time-varying DGGS-based airspace model, and if the computational evaluation demonstrates how the selected efficiency strategies affect the practicality of generating and using the model.

3.2 Derivation of location-based airspace metrics

The third research gap motivates the second research question, corresponding hypothesis, and experiment. The gap concerns the close coupling between many existing airspace representations and the single analytical purpose for which they were developed. The unresolved issue is whether a reusable, multi-source 4D airspace model can instead serve as a common analytical foundation from which multiple derived analytical products can subsequently be generated and interpreted. Whereas Experiment #1 investigates the construction and computational practicality of the base airspace model, Experiment #2 investigates its subsequent analytical reuse.

A reusable airspace model could potentially support several types of derived analyses such as traffic-density, congestion, environmental, operational, and safety-related metrics. The underlying methodological question is therefore intentionally formulated independently of a particular spatial framework or analytical application. It concerns how heterogeneous, time-varying observations can be transformed into multiple derived quantities that share a common spatial and temporal reference. This leads to the second research question:

Research question #2

How can a reusable 4D airspace model be used to derive and interpret multiple types of location-based airspace metrics from heterogeneous, time-varying airspace data contained within it?

Because a single experiment cannot evaluate every possible airspace metric, Experiment #2 investigates this general question through a location-based airspace risk case study. Risk analysis provides a demanding test of the model's analytical capability because it requires quantities derived from two heterogeneous, time-varying data sources to be computed at common three-dimensional locations and times. It also requires the individual contributions to be examined separately and combined into a coherent analytical product.

The proposed experiment is to use the DGGS-based structure developed in Experiment #1. Aircraft and weather observations are associated with common DGGS cells and time steps, after which application-specific computations can derive quantities associated with those same locations and times. Aircraft-proximity and hazardous weather components can therefore be computed separately and subsequently combined into a holistic location-based airspace risk metric without redesigning the underlying airspace representation. The second hypothesis is:

Hypothesis #2

A reusable DGGS-based 4D airspace model can support multiple co-indexed location-based analytical products. A layered risk analysis case study using aircraft proximity risk, hazardous weather risk, and their combined effects can demonstrate this capability.

Research Experiment #2 tests Hypothesis #2 using the DGGS-based airspace model developed in Experiment #1. Its inputs are the processed aircraft, weather-radar, and storm cell datasets associated with DGGS cells and time steps. Its outputs are three related location-based analytical products: a midair-collision risk metric derived from aircraft motion and spatial convergence, a weather risk metric derived from hazardous weather observations, and a combined risk metric derived from the two individual components. Time-series plots, three-dimensional risk maps, and 2D horizontal risk maps are used to examine and interpret the resulting spatial and temporal patterns.

The risk formulations are developed specifically to evaluate whether heterogeneous observations represented within the common airspace model can be transformed into separate and combined analytical products. The experiment provides a controlled case study through which the broader capability identified in Research Question #2 can be evaluated.

The experiment is considered successful if:

1. the aircraft proximity and hazardous weather metrics can be computed at common DGGS cells and time steps;
2. the individual metrics exhibit interpretable spatial and temporal variation consistent with their respective aircraft and weather inputs;
3. the two components can be combined within the same location-based analytical field; and
4. the resulting products can be interpreted using time-series, three-dimensional, and two-dimensional visualizations.

The resulting metrics are treated as analytical indicators developed for the validation objectives of this thesis. They are not presented as calibrated operational collision probabilities, certified meteorological products, or deployable safety-assessment models.

3.3 Behaviour-based aircraft categorization using machine learning

The fourth research gap motivates the third research question, corresponding hypothesis, and experiment. This gap concerns the identification and interpretation of recurring operational structure in airspace model data when the relevant groupings are not known in advance. Existing approaches can require categories, rules, labels, or analytical boundaries to be defined through subject matter expertise or application-specific assumptions before the underlying structure of the data has been examined. The underlying methodological question is therefore how recurring operational structure can be identified primarily from the observed airspace model data themselves, while domain knowledge is subsequently used to support interpretation and validation. This leads to the third research question:

Research question #3

How can trajectory-level information from a reusable 4D airspace model be used to identify recurring operationally meaningful patterns in a data-driven manner and support their interpretation when the relevant groupings are not defined in advance?

This thesis proposes unsupervised machine learning clustering as a solution to this problem. Unsupervised clustering is suited to the intended application because it can identify structure directly from unlabelled trajectory data without requiring behavioural categories to be specified before analysis. Complete aircraft trajectories are treated as the objects being grouped, allowing recurring spatial and temporal movement patterns to determine the initial cluster structure.

The resulting trajectory groups are subsequently interpreted as behaviour-based aircraft categories using their spatial and kinematic characteristics. In this formulation, aircraft type, airworthiness category, runway identity, and expert-defined operational labels are excluded from the clustering inputs and are introduced only after the groups have been generated to support interpretation and independent comparison. The proposed approach therefore shifts the role of subject-matter expertise from defining the categories a priori to interpreting and validating patterns identified directly from the observed data.

The third hypothesis is therefore:

Hypothesis #3

Unsupervised clustering applied to trajectory-level outputs from a reusable 4D airspace model can identify recurring aircraft operational patterns directly from observed trajectory structure without predefined behavioural labels, and the resulting data-derived groups can subsequently be interpreted as operationally meaningful behaviour categories using post-clustering spatial, kinematic, and domain evidence.

Research Experiment #3 tests Hypothesis #3 using trajectory-level outputs extracted from the DGGS-based airspace model developed in Experiment #1. The trajectories are transformed into a common fixed-length representation suitable for unsupervised analysis. A supporting comparative evaluation of DBSCAN, HDBSCAN, KMeans, and GMM is used to select the clustering implementation for the aircraft categorization case study, with KMeans selected as the most suitable tested approach for the intended application.

The KMeans implementation is applied to identify recurring trajectory groups within the airspace-model outputs. The resulting groups are characterized using trajectory geometry and associated spatial and kinematic variables, including altitude, heading, vertical rate, and velocity. Once the clustering is complete, the experiment then evaluates how the level of category granularity affects the operational structure represented by the groups and compares the behaviour-based categories with independent aircraft-reference information.

The experiment is considered successful if:

1. recurring trajectory groups are identified directly from the airspace-model outputs without predefined behavioural category labels;
2. the resulting groups exhibit distinct spatial and kinematic characteristics that support their interpretation as operationally meaningful aircraft behaviour categories;
3. variation of clustering granularity reveals coherent finer operational subpatterns within the broader identified behaviours; and
4. independent aircraft-reference information supports post hoc interpretation and demonstrates how the data-driven behaviour categories relate to conventional predefined aircraft classifications.

Chapter 4: Methodology

This chapter presents the methodology developed to integrate historical aerospace data into a DGGS-based 4D airspace model and to use this model for subsequent analytical applications. The methodology is organized around seven main components: 1) airspace entity and data source selection, 2) data collection and preprocessing, 3) computational execution, 4) DGGS-based airspace model construction, 5) location-based airspace metric computation, 6) machine learning analysis, and 7) visualization and interpretation tools.

Figure 18 summarizes the proposed methodology. Historical aerospace data are first selected, collected, processed, and structured into a DGGS-based 4D airspace model. This model then supports two downstream analytical branches. The first branch computes location-based airspace metrics, with risk analysis used as the case study presented in this thesis. The second branch applies unsupervised machine learning. The case study presented in this thesis uses a clustering algorithm to identify recurring structure in the airspace model output. Visualization and interpretation tools are used throughout the methodology to verify intermediate outputs and support interpretation of the results.

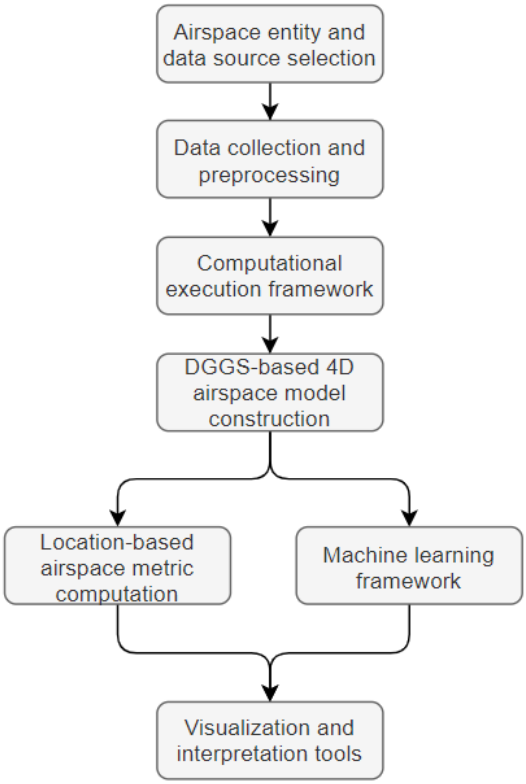


Figure 18: Overview of the methodology

Figure 19 shows the relationship between the methodology and research experiments. The figure lays out which parts of the methodology are validated by the three research experiments.

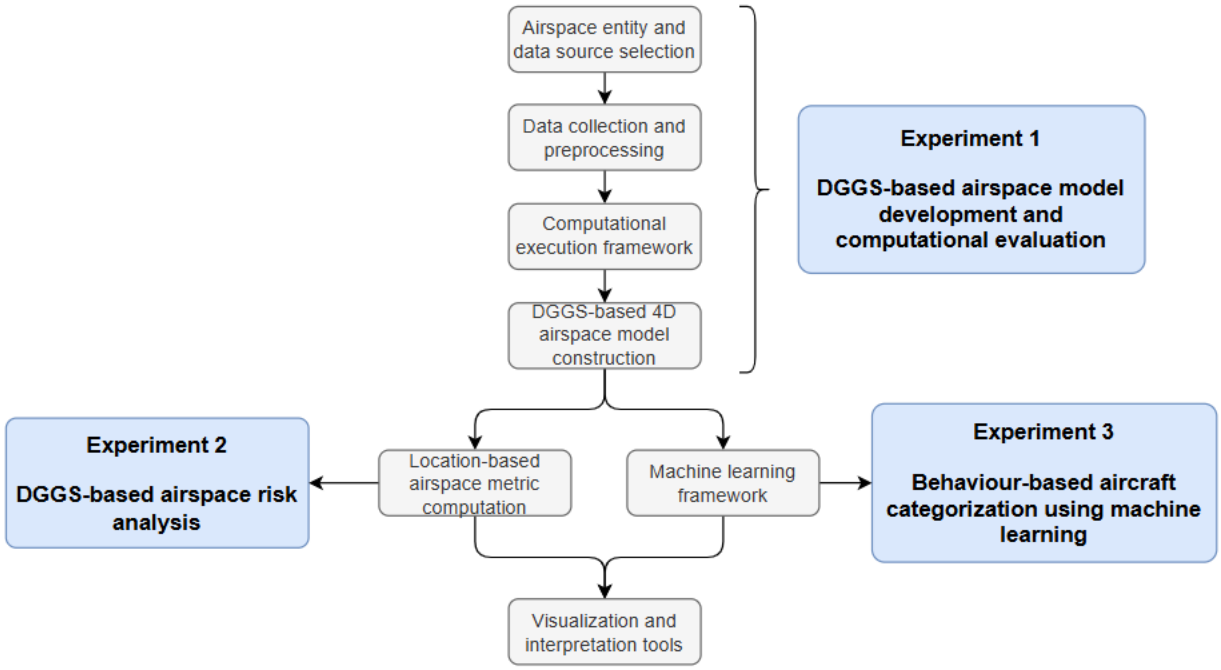


Figure 19: Relationship between methodology and research experiments

4.1 Airspace entity and data source selection

The methodology begins with the selection of the airspace entities and data sources suitable for the intended analysis. An airspace entity is any object or spatially distributed quantity that can be represented by a 3D location and time. In this thesis, the primary airspace entities are aircraft trajectories, weather radar observations, and storm cell representations derived from the radar data.

The selected entities must satisfy two requirements. First, they must contain sufficient spatial and temporal information to be mapped into the airspace model. Second, they must be relevant to at least one analytical objective of the thesis. Aircraft trajectory data are required for modelling air traffic patterns, calculating aircraft collision risk metrics, and implementing trajectory-level clustering. Weather radar and storm cell data are required for modelling hazardous weather and extracting combined air traffic-weather risk metrics.

The methodology is not restricted to these entities. Other airspace-related datasets, such as terrain, ground infrastructure, route structures, emissions fields, or future UAS operations data, could be incorporated if they contain suitable geospatial and temporal information. However, the thesis focuses on aircraft and weather data because they provide two distinct types of time-varying aerospace data and directly support the experimental validation described in Chapters 5, 6, and 7.

4.2 Data collection and preprocessing

During the data collection and preprocessing stage, raw aerospace data is converted into analysis-ready datasets that can be mapped into the DGGS-based airspace model. The objective of this stage is to integrate different datasets into a common spatial, temporal, and computational structure so that they can be used by the airspace model. The required operations depend on the data source, with the main steps being data collection, coordinate transformation, spatial and temporal filtering, temporal alignment, interpolation, and smoothing where required, and outlier removal as presented in Figure 20. Chapter 5 presents the implementation of this methodology specifically for air traffic and weather data.

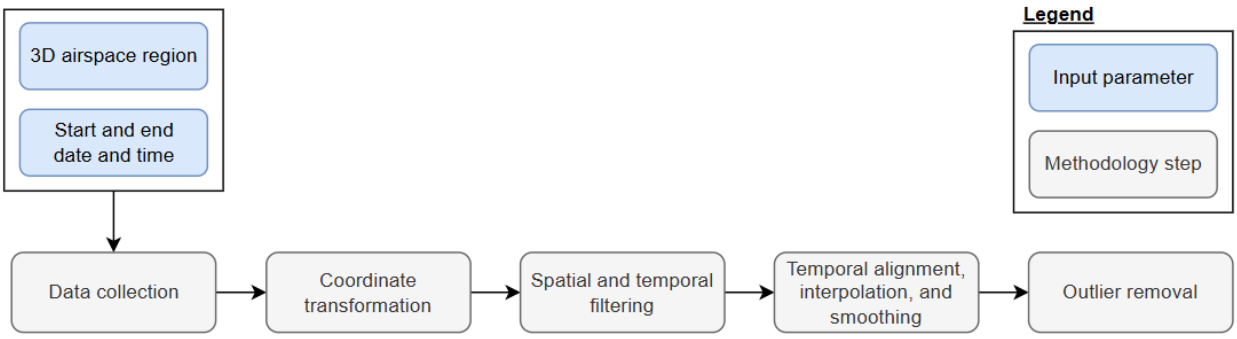


Figure 20: Main steps for data collection and processing

4.2.1 Data collection

The raw data is imported from its source repository based on the relevant 3D airspace region and time segment.

4.2.2 Coordinate transformation

Geospatial datasets are transformed into a common coordinate system before being mapped into the airspace model. A common spatial reference is required because the methodology combines datasets that may originally be stored using different coordinate systems, units, and geographic conventions.

For the local and regional airspace domains considered in this thesis, horizontal position is transformed to the Universal Transverse Mercator (UTM) coordinate system [449]. UTM represents projected position in meters using Easting and Northing coordinates. These coordinates are used as the x- and y-axes, respectively, while geometric altitude is retained as the vertical z-axis. The airspace model therefore represents three-dimensional position using the Cartesian coordinate tuple $(\mathrm{E}, \mathrm{N}, \mathrm{z})$, corresponding to Easting, Northing, and altitude. This representation allows distances, cell boundaries, and DGGS dimensions to be calculated directly in metric units. Longitude and latitude are retained where useful for geographic referencing, mapping, and visualization, while the main DGGS construction and spatial calculations use the projected Easting and Northing coordinates together with altitude. The specific UTM transformation implemented for the experimental datasets is described in Section 5.2.2.2.

4.2.3 Spatial and temporal filtering

Data filtering is applied to retain only observations relevant to the selected airspace region, altitude range, and time interval. This step removes observations outside the model domain and reduces the amount of data carried through the processing pipeline. Depending on the data source, additional filtering may be applied using physically meaningful thresholds, such as altitude limits, radar reflectivity thresholds, or minimum trajectory quality requirements.

4.2.4 Temporal alignment, interpolation, and smoothing

Time-varying aerospace datasets often contain irregular sampling intervals, missing observations, or use different update rates. Where required, the data are aligned to a common time basis before being used by the airspace model. Aircraft trajectories may require smoothing or interpolation to reduce the effect of irregular ADS-B sampling and missing records. Weather radar data may instead be aligned by radar volume time or storm cell time step.

Interpolation and smoothing are applied only when they support the intended analysis and preserve the physical meaning of the data. The specific procedure depends on the data source and is described in Experiment #1 in Chapter 5.

4.2.5 Outlier removal

Invalid or physically unrealistic observations are removed after spatial filtering and time alignment. The outlier criteria are data-source specific. For aircraft data, outliers may include unrealistic positions, altitudes, velocities, or discontinuities in a trajectory. For weather data, outliers may include invalid radar values, missing fields, or observations outside the relevant analysis threshold. The objective is to remove observations that would distort the airspace model while preserving valid low-density or uncommon operational behaviours.

4.3 Computational execution framework

The methodology includes a computational execution framework as a fundamental part of the airspace model to enable it to remain computationally efficient and practical for use on large airspace datasets. This framework controls how data are processed, how intermediate outputs are stored, and how previously generated results are reused.

Airspace models can be generated using the methodology for two distinct execution modes. In BUILD mode, the required datasets and model outputs are regenerated completely from source or raw data. This is the mode used to gather new data into the model. In LOAD mode, previously generated outputs that match the selected configuration are loaded from local storage. This distinction separates the cost of constructing the airspace model from the cost of reusing an existing model.

The framework also supports feature selection and dataset partitioning. Feature selection reduces the number of variables carried through the processing workflow by retaining only the ones essential for the intended analysis. Dataset partitioning divides large time-varying datasets into smaller temporal subsets that are processed independently and recombined after processing. These strategies are intended to improve computational practicality without changing the intended use case of the airspace model.

At this stage of the methodology, computational performance is evaluated using metrics such as wall-clock time, stage-level runtime, memory use, throughput, and relative speedup. These metrics are used in Experiment #1 to assess how the modelling workflow behaves under selected combinations of execution strategies and airspace configurations.

4.4 DGGS-based 4D airspace model construction

The DGGS-based 4D airspace model is the central structure of the proposed methodology. Its purpose is to organize processed aerospace data into a common spatial and temporal framework. Once aircraft, weather, or other airspace entities are represented in this framework, they can be compared, combined, queried, visualized, and used for analytical tasks.

The model is described as four-dimensional because it represents three-dimensional airspace and time. The three spatial dimensions are represented by a cartesian coordinates system with x, y, and z axes, where altitude is measured on the vertical z-axis. Time is retained so that each airspace entity can be associated with both a spatial location and a time step. This allows moving aircraft, evolving weather systems, and storm cell objects to be represented in the same airspace model even though they differ in format, update rate, and physical meaning.

The DGGS model is constructed by dividing the selected airspace volume into discrete three-dimensional cells. Each cell has 3D spatial bounds, a centroid location, and a unique cell identifier. The cell identifier allows different datasets to be linked to the same spatial structure. The centroid can be used for cell-level calculations, such as risk metrics or spatial summaries. The cell bounds determine which observations belong to each cell.

The methodology uses a DGGS-based implementation for bounded airspace regions rather than for global Earth-scale representation. In this thesis, the model is applied to local and regional airspace volumes, such as terminal-area airspace around a single airport or metropolitan airspace containing multiple airports. UTM coordinates are used because these bounded regions can be represented accurately in a metric cartesian coordinate system. This allows cell dimensions, distances, centroids, and spatial boundaries to be computed directly in metres. The modelled region is defined using equal east–west and north–south extents, producing a square horizontal footprint. This treats both horizontal dimensions consistently and simplifies grid construction and spatial indexing. The vertical extent is specified independently using lower and upper altitude bounds because the operationally relevant altitude range is not necessarily equal to the horizontal extent and may differ according to the flight levels represented. Consequently, the complete airspace volume is generally a square-based cuboid, although it becomes a cube when the horizontal and vertical extents are equal. The volume is subdivided into cubic DGGS cells at each resolution level.

The grid *resolution* controls the spatial detail of the model. A finer resolution provides greater local detail but increases the number of cells and the computational cost. A coarser resolution reduces the number of cells but may hide important local structure. The methodology can therefore use either a uniform resolution or a mixed-resolution structure. In a mixed-resolution structure, selected regions of interest are represented with smaller cells while the surrounding airspace is represented with larger cells. This allows the model to preserve detail in important areas, such as airport regions, while limiting unnecessary computational cost elsewhere.

After the grid is created, processed airspace data are mapped to DGGS cells. Each datapoint is assigned to a cell using its horizontal UTM coordinates and altitude. Its time value is retained so that the mapped dataset represents both spatial position and temporal evolution. This creates a common 4D representation in which aircraft trajectory points, radar observations, and storm cell objects can be stored using the same cell-based structure.

The mapped data can be used at two levels. At the observation level, individual aircraft positions or weather observations are retained with their associated DGGS cell identifiers and times. This is useful for trajectory analysis, visualization, and machine learning applications. At the cell level, observations can be aggregated within cells and time steps to compute summary values, such as counts, maximum reflectivity, density, or risk metrics. This is useful for location-based airspace analysis.

Spatial indexing is used to organize the DGGS cell structure. Each grid cell receives a unique centroid identifier and a Morton code, with the latter obtained by interleaving the integer indices of its three spatial axes. Morton ordering places many spatially adjacent cells near one another in a one-dimensional sequence, supporting spatial ordering and computational batching. The mapping stage assigns processed observations to DGGS cells by calculating their cell indices from projected coordinates, identifying the appropriate grid resolution, and verifying the corresponding cell bounds.

The model is stored as a collection of related data artifacts. The DGGS grid contains cell bounds, centroids, resolution levels, and identifiers, while the processed aircraft, radar, and storm datasets retain their respective observation coordinates, timestamps, source-specific attributes, and associated DGGS identifiers. Subsequent analytical stages generate additional datasets containing derived results. This

organization preserves the individual observations and their relationships while allowing data to be associated with common spatial and temporal references. During analysis, the relevant persisted datasets are loaded into tabular in-memory structures, where observations can be selected and processed according to the requirements of each application.

A more detailed theoretical background on implementing a DGGS framework, which served as the basis for the overview presented in this section, can be found in Appendix A.

The outputs of this methodology component are reusable DGGS-based airspace model products. These include the DGGS grid, cell identifiers, cell centroids, cell bounds, spatial indices, and processed airspace datasets mapped to the DGGS structure. These outputs form the common technical foundation for the rest of the thesis. Chapter 5 evaluates the construction and computational practicality of the model. Chapter 6 uses the model to compute location-based risk metrics. Chapter 7 uses airspace model outputs for machine learning-based aircraft categorization.

4.5 Location-based airspace metric computation

The DGGS-based 4D airspace model can be used to compute location-based airspace metrics. A location-based airspace metric is a quantitative value associated with a DGGS cell and a time step. These metrics are computed from the airspace entities mapped to the model and can represent operational, safety, environmental, or traffic-related quantities, depending on the model's intended use case.

In this thesis, the metric computation branch of the methodology is demonstrated in Chapter 6 using an airspace risk analysis use case. Aircraft collision risk, weather-related risk, and combined air traffic-weather risk are computed from the DGGS-structured aircraft and weather data. These risk metrics are used to test whether the airspace model can generate interpretable analytical outputs rather than only store or visualize historical data.

The inputs to this component are the processed airspace datasets mapped to DGGS cells, the relevant model metadata, and the parameters required for the selected metric. These parameters may include safety thresholds, time windows, weather thresholds, or other case-specific values. The computation then produces a new spatiotemporal dataset in which the calculated metric is associated with selected DGGS cells and time steps.

The specific metric equations depend on the application. For the case study presented in Chapter 6, the methodology is applied to airspace risk analysis. In other applications, the same structure could support metrics for traffic density and air traffic management, air emissions studies, airspace design, or anomaly detection.

4.6 Unsupervised machine learning framework

The machine learning component of the methodology is used to identify recurring structure in airspace model outputs. In this thesis, this component is implemented using unsupervised clustering as a means of identifying patterns without relying on predefined labels. Aircraft trajectory categorization is used as the case study application presented in Chapter 7.

The clustering workflow begins by selecting the airspace model outputs for the target application. In the case of aircraft trajectory categorization, the input data consist of trajectory-level information extracted from processed ADS-B data. Each trajectory is represented using a fixed-length feature representation that captures spatial and temporal behaviour. The representation can include position, time-normalized trajectory shape, duration, and other operational variables where relevant.

Candidate clustering models are then trained and evaluated. The methodology considers both quantitative clustering metrics and qualitative operational interpretation. Quantitative metrics provide information on cluster separation and compactness. Qualitative interpretation evaluates whether the resulting clusters correspond to meaningful operational behaviour patterns when visualized and compared with airspace knowledge or available aircraft reference information.

The final clustering output is a trajectory-level assignment in which each trajectory is associated with a cluster identifier. These assignments can then be used to examine aircraft behaviour categories, compare clusters with reference categories, and interpret recurring operational structure in the airspace model outputs. At the methodology level, this component requires only that airspace-model outputs be transformable into an analysis-ready representation suitable for unsupervised machine learning. The choice of clustering algorithm, trajectory representation, hyperparameters, evaluation metrics, and interpretation procedure is application-dependent and is therefore defined and evaluated as part of Experiment #3 in Chapter 7.

4.7 Visualization and interpretation tools

Visualization and interpretation tools are used throughout the methodology to verify data processing, inspect the DGGS-based airspace model, interpret metric outputs, and evaluate clustering results. The methodology produces high-dimensional outputs that combine 2D position, altitude, time, multiple airspace entities, computed metrics, and cluster assignments. These tools are therefore required to reveal spatial, temporal, and multivariate relationships that would be difficult to identify or interpret from tabular outputs alone.

The visualization tools support all three research experiments. In Experiment #1, they are used to verify the spatial and temporal structure of the DGGS-based airspace model and to inspect processed aircraft and weather data. In Experiment #2, they are used to interpret aircraft collision, weather-related and combined risks across 3D space and time. In Experiment #3, they are used to assess if clustering results correspond to interpretable aircraft trajectory patterns.

The tools developed as part of this thesis include two-dimensional geospatial plots, three-dimensional airspace plots, time-varying trajectory visualizations, DGGS grid plots, risk heatmaps, aligned trajectory-profile plots, and clustering evaluation plots. These tools are not treated as a separate research experiment in the final thesis structure. Instead, they are treated as a cross-cutting methodological component that supports validation, interpretation, and communication of results.

4.8 Methodology implementation

The methodology is implemented in Python using a modular pipeline that can run individual components of the methodology. Table 11 summarizes the main Python files used in the implementation of the methodology.

The central execution script is *run_simulation.py*, which coordinates and runs the data processing, DGGS-based airspace model generation, airspace metric computation, machine learning model, and visualization stages. The Python pipeline is controlled primarily through *configuration.py*, which defines the airspace region, time interval, grid properties, data sources, execution modes, and visualization settings used to choose which parts of the methodology to run.

The modular structure supports reproducibility because each stage produces intermediate outputs that can be saved, loaded, inspected, and reused. This allows a single airspace model configuration to support

multiple analyses without regenerating the source data each time. It also allows computational performance to be measured at different stages of the workflow.

Table 11: Summary of the main Python files used in the implementation of the methodology

File	Role in methodology
configuration.py	Defines model configuration, input data settings, execution modes, and experiment switches.
run_simulation.py	Central execution script that coordinates the full methodology.
data_processing.py	Collects, loads, filters, processes, and partitions ADS-B and weather data.
geospatial_model_generator.py	Builds or loads the DGGS grid and maps airspace data to DGGS cells.
risk_model.py	Computes aircraft collision, weather, and combined risk metrics.
clustering_model.py	Builds trajectory-level feature representations and applies unsupervised clustering.
visuals_formatting.py	Generates reusable visualization outputs for model verification, risk interpretation, and clustering analysis.
generic_functions.py	Provides shared geospatial, temporal, I/O, and performance-tracking utilities.

Figure 21 shows the relationships between the main Python files used in the implementation of the methodology.

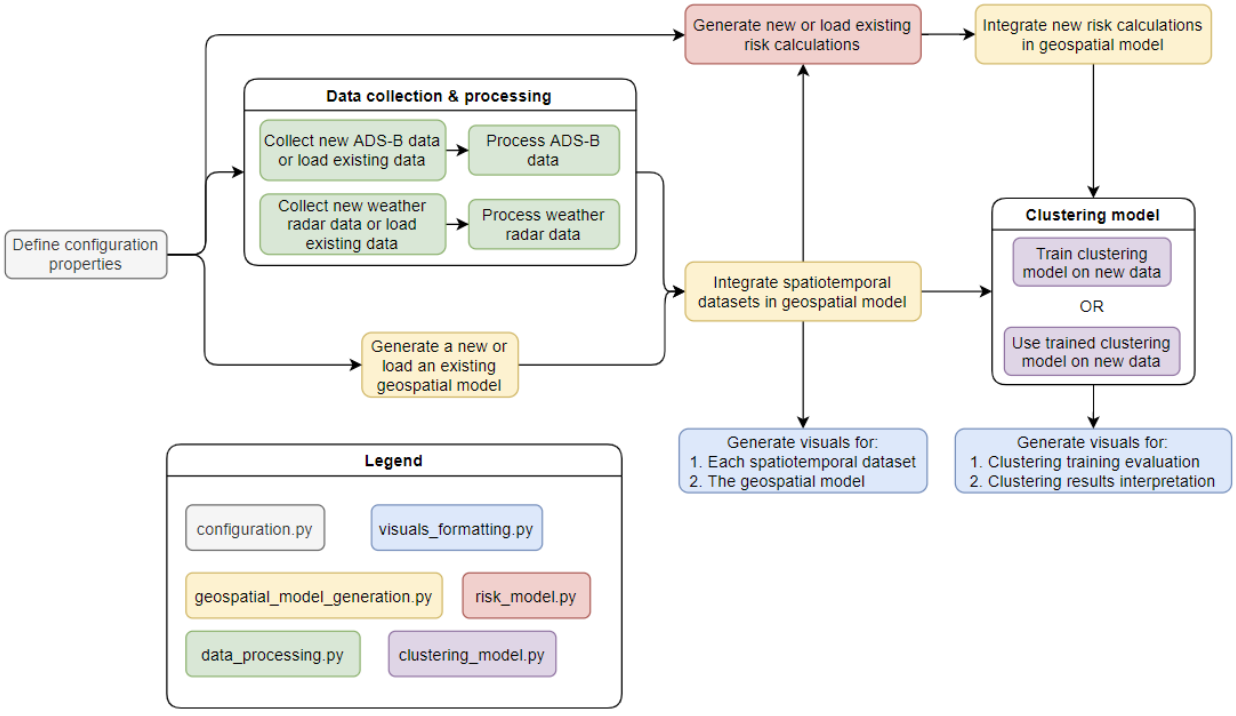


Figure 21: Functional workflow diagram of the Python code

Chapter 5: DGGS-based airspace model development and evaluation

This chapter presents Experiment #1, which describes the development of the DGGS-based 4D airspace model and the evaluation of its computational behaviour. As defined in Chapter 3, Experiment #1 addresses Research Gaps 1 and 2 and tests Hypothesis #1: first, by evaluating whether heterogeneous historical aerospace data can be integrated into a common, reusable DGGS-based 4D airspace model; and second, by evaluating whether feature selection and dataset partitioning can improve the computational practicality of the modelling workflow while preserving the model outputs required by downstream analyses.

Parts of the airspace-modelling methodology evaluated in this chapter were introduced in earlier conference and journal publications by the author. Vincent-Boulay et al. [6] presented a DGGS-based geospatial airspace-modelling approach for integrating ADS-B air traffic and weather-radar data within a common spatial representation. The framework was subsequently extended by Vincent-Boulay and Marsden [2], who used a DGGS-based geospatial airspace model as the foundation for location-based airspace risk analysis. Experiment #1 develops these ideas further within the unified thesis methodology by formalizing the reusable 4D airspace-model representation and evaluating its computational generation and reuse across the two model configurations.

Two representative airspace models are used to test the methodology across different spatial, temporal, and resolution settings, as well as data conditions. The resulting models are first examined to verify that aircraft and weather data are correctly represented within the common DGGS-based structure. Their computational behaviour is then evaluated using alternative feature-selection and dataset-partitioning strategies in BUILD and LOAD execution modes. Together, these analyses provide the experimental evidence used to answer Research Question #1 and to evaluate Hypothesis #1.

This chapter is organized as follows. Section 5.1 defines the experimental design, the two airspace model test cases, the computational scenarios, and the evaluation measures. Section 5.2 describes the implementation of the DGGS-based airspace model. Section 5.3 presents the resulting airspace model outputs and uses visualization to verify their spatial and temporal properties. Section 5.4 evaluates the computational behaviour of the modelling workflow under the selected execution strategies. Section 5.5 discusses the combined results with respect to Research Question #1 and Hypothesis #1.

5.1 Experimental design, inputs, and outputs

5.1.1 Experiment objective and validation criteria

Experiment #1 evaluates two linked aspects of Hypothesis #1. The first evaluates whether heterogeneous historical aerospace datasets can be transformed into a common DGGS-based 4D airspace representation. Evidence for this part of the hypothesis is provided by the successful processing, georeferencing, DGGS mapping, temporal indexing, storage, and visualization of historical ADS-B aircraft trajectory and NOAA NEXRAD weather radar data using two representative airspace model configurations.

The second evaluates the computational practicality of generating and reusing these airspace models. Feature selection and dataset partitioning are evaluated as alternative processing strategies, while BUILD and LOAD represent two execution modes corresponding respectively to model generation and model reuse. The computational comparison focuses on execution time and memory demand. A strategy is considered computationally beneficial only when any reduction in computational cost is achieved while preserving the model information required by subsequent analytical applications.

Table 12 summarizes the inputs, outputs, and validation criteria used for the two parts of Experiment #1.

Table 12: Summary of inputs and outputs to the airspace model development and evaluation experiment

Evaluation objective	Inputs	Outputs	Validation criterion
Airspace model development	• Historical ADS-B aircraft trajectory data; • NOAA NEXRAD weather radar data; • aircraft reference metadata where needed for interpretation; • model parameters defining spatial extent, altitude range, temporal interval, DGGS resolution, and refinement regions	• DGGS grid; • processed and georeferenced aircraft trajectory dataset; • processed and georeferenced radar and storm cell dataset; • DGGS cell identifiers; • Morton-code spatial indices; • model verification figures	Demonstrate that multiple historical aerospace datasets can be integrated into a common 3D, time-indexed DGGS-based airspace model
Computational evaluation	Computational execution parameter settings including different feature selection, dataset partitioning, and execution modes.	• Total wall-clock time; • mean RSS memory usage; • peak RSS memory usage; • dataset-size comparison tables; • computational evaluation figures	Determine how FS and DP alter computational cost while preserving required model outputs

5.1.2 Airspace model test cases

Experiment #1 uses two representative DGGS-based airspace models to evaluate the model-development component of the methodology. Both models are generated using the same modelling workflow and integrate historical ADS-B aircraft trajectory data and NOAA NEXRAD weather radar data. They differ in spatial extent, temporal duration, DGGS resolution strategy, and dataset composition. The two models therefore provide complementary test cases for evaluating whether the methodology can represent heterogeneous historical airspace data under different spatial and resolution conditions.

Table 13 summarizes the properties of the two airspace models.

Table 13: Airspace model properties used in Experiment #1

Configuration parameter	Airspace model #1 properties	Airspace model #2 properties
Data type	• ADS-B aircraft trajectory data; • NOAA NEXRAD weather radar data	• ADS-B aircraft trajectory data; • NOAA NEXRAD weather radar data
Date and time	22 Nov. 2022, 23:00 UTC – 23 Nov. 2022, 04:00 UTC (5 h)	21 Nov. 2022, 23:00 UTC – 23 Nov. 2022, 23:00 UTC (48 h)
Model location	Terminal-area airspace centred on LGA	Greater New York metropolitan airspace including 4 airports
Model boundaries	Smaller region: 5 x 5 x 5 km volume	Larger region: 50 x 50 x 14 km volume
DGGS resolution	Uniform high resolution 100 m cells	Mixed resolution of coarse 5 km cells for the base cells and fine 2.5 km cells near airports
Traffic characteristics	Dense terminal-area operations near LGA	Mixed-density regional operations involving multiple airports
Aircraft categories observed	Multiple aircraft types including small, medium, and large fixed wing aircraft and rotorcraft	Multiple aircraft types including small, medium, and large fixed wing aircraft and rotorcraft
Primary validation purpose	Test high-resolution local representation	Test larger-domain and mixed-resolution representation

The two models were selected to provide complementary validation conditions for Experiment #1. Airspace model #1 tests whether the methodology can represent dense terminal-area operations at high spatial resolution. Airspace model #2 tests whether the same methodology can represent a larger regional airspace using a mixed-resolution DGGS structure. Together, the two models evaluate whether aircraft and weather data can be integrated into a common 4D DGGS framework across different model sizes, spatial scales, data densities, and resolution strategies.

The New York metropolitan airspace was selected for both models because it provides a complex operational environment for evaluating the methodology. The region contains several major airports in close proximity, including JFK, EWR, LGA, and TEB, and includes a mixture of commercial air transport, business aviation, general aviation, and rotorcraft operations. This makes it suitable for testing whether the model can represent multiple aircraft categories, dense traffic patterns, airport-centred operations, and broader regional flows within the same methodological framework.

Airspace model #1 represents a 5 km × 5 km × 5 km terminal-area volume centred on LaGuardia Airport. It uses a uniform DGGS cell width of 100 m and spans a five-hour temporal interval. The fine spatial resolution is chosen to preserve detailed terminal-area variation within a compact domain and to provide the location-based spatial framework used in the risk-analysis case study in Experiment #2. Airspace model #1 therefore emphasizes local spatial detail and temporal continuity, whereas Airspace model #2 evaluates the methodology over a substantially larger regional domain using a mixed-resolution DGGS structure.

Airspace model #2 represents a 50 km × 50 km × 14 km regional volume covering the greater New York metropolitan airspace. It uses a mixed-resolution DGGS structure, with 5000 m base cells and 2500 m refined cells around selected airport regions. This model is intended to test whether the same methodology can support a larger spatial domain while retaining local refinement where greater spatial detail is required. It emphasizes spatial scale, multi-airport structure, mixed traffic density, and DGGS resolution hierarchy.

The altitude dimension of Airspace model #2 is set at 14 km because almost no air traffic operates above this altitude.

Figures 22-25 show the DGGS domains for the two airspace models. Figures 22 and 23 show the 3D and 2D DGGS structure of Airspace model #1, repectively, while Figures 25 and 26 show the corresponding structure for Airspace model #2. The blue cells represent the base DGGS resolution in each model. For Airspace model #2, the red cells represent refined DGGS cells around airport regions. These figures show how the same DGGS-based methodology can represent both a compact high-resolution terminal model and a larger mixed-resolution regional model.

The two airspace models are used to evaluate the model-development component of Experiment #1. Together, they test whether the same methodology can integrate historical aircraft and weather data into reusable 4D DGGS representations under different spatial extents and DGGS resolution strategies. The computational evaluation is defined separately in the following section using the fixed spatial configuration of Airspace Model #2 and for six different time ranges.

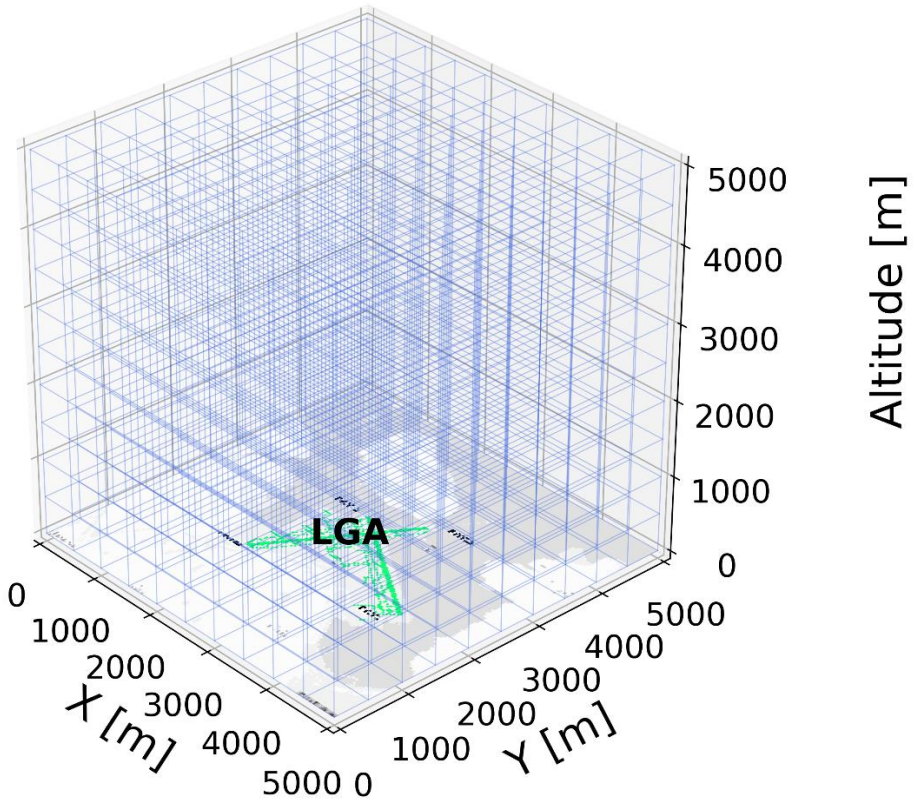


Figure 22: 3D DGGS structure of Airspace model #1

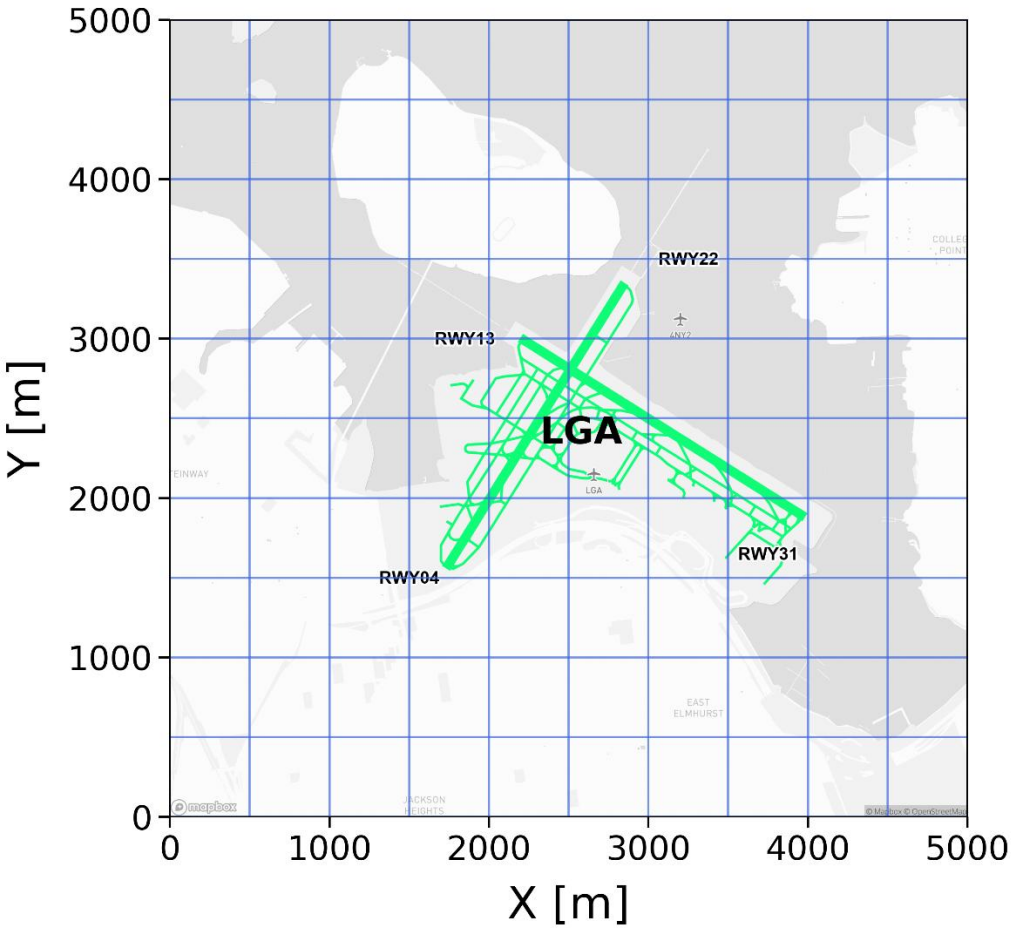


Figure 23: 2D view of Airspace model #1 showing LaGuardia airport

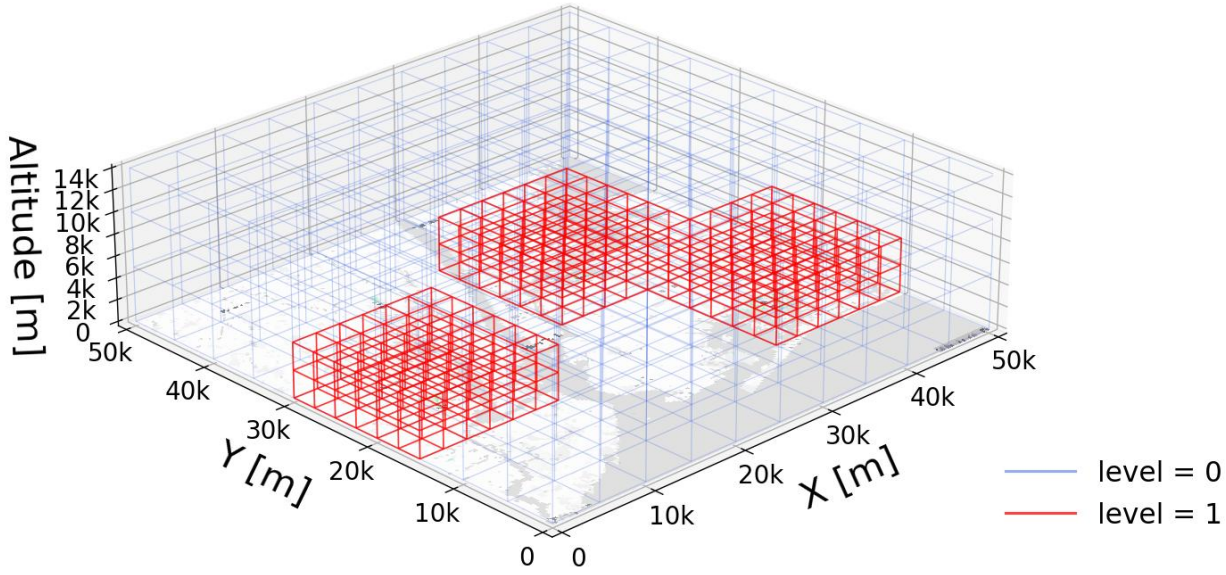


Figure 24: 3D DGGS structure of Airspace model #2

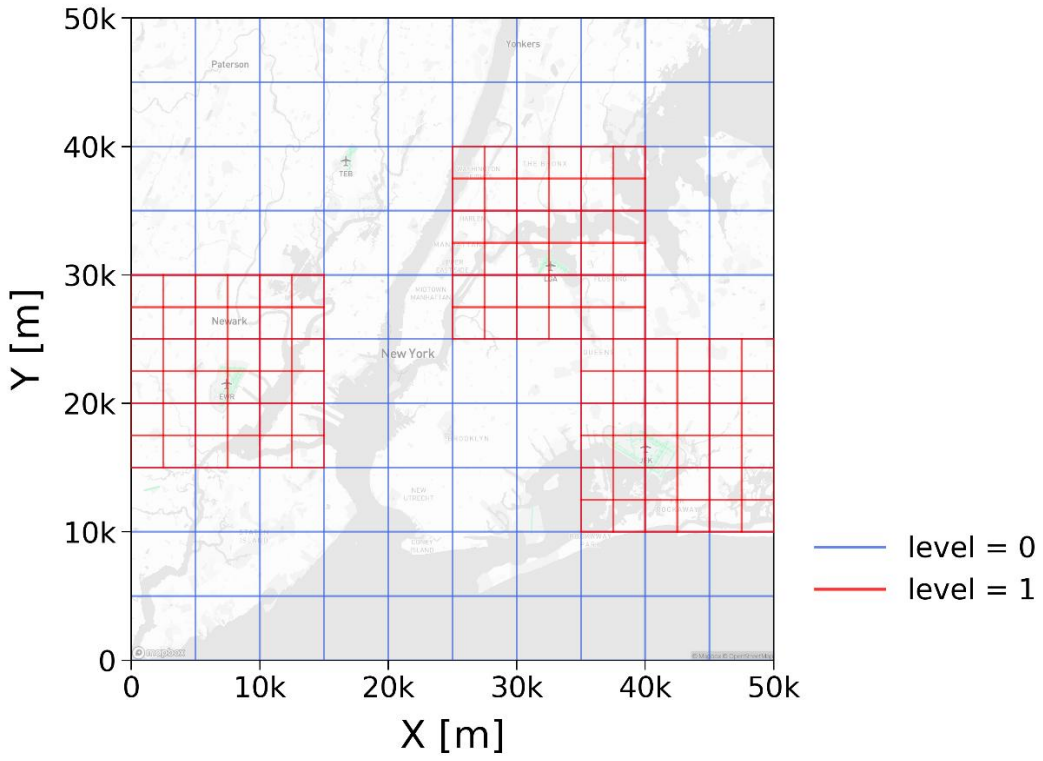


Figure 25: 2D view of Airspace model #2 showing major airport locations

5.1.3 Computational workload evaluation design

The computational evaluation is separated from the two model-development test cases described above. Rather than comparing the computational workloads associated with Airspace model #1 and Airspace model #2, the computational study fixes the spatial extent, altitude range, and DGGS resolution structure of Airspace model #2 and varies the length of the historical data interval, and consequently the amount and composition of the source data processed. This design allows the relationship between increasing historical data workloads and computational performance to be evaluated while keeping the spatial extent and DGGS configuration fixed.

Six nested historical time ranges (TR1–TR6) are evaluated and summarized in Table 14. They are nested in the sense that TR1-TR5 are included in TR6, and TR1-TR4 are included in TR5, and so on. TR1 represents 1 h of historical data, followed by TR2 = 3 h, TR3 = 6 h, TR4 = 12 h, TR5 = 24 h, and TR6 = 48 h. All time ranges begin at 23:00 UTC (18:00 EST) on 21 November 2022, such that each longer range contains the complete interval represented by the preceding ranges. The combined size of the selected raw ADS-B and radar source files increases from 0.540 GiB for TR1 to 17.716 GiB for TR6. The six time ranges therefore define progressively larger computational workloads while keeping the spatial extent and DGGS configuration fixed. These source-data sizes describe the files read during BUILD execution; they do not represent the amount of data simultaneously resident in memory or the size of the processed model outputs subsequently read during LOAD execution.

Historical duration and raw source-data size describe the nominal scale of each time range, but they do not fully describe the amount or type of information processed. In this thesis, **dataset composition** refers to the quantities of aircraft datapoints, trajectory segments, radar observations, and derived weather information contained within a historical interval. These quantities vary with actual aircraft and weather activity and therefore do not increase proportionally from TR1 to TR6. Dataset composition is consequently considered alongside duration and source-data size when interpreting computational runtime and memory demand.

Table 14: Historical time-range and dataset characteristics

Time range	Duration	Total raw source-data size (GiB)	ADS-B data			Radar data		
			ADS-B raw source-data size (GiB)	Datapoints	Trajectories	Radar raw source-data size (GiB)	Radar points	Radar volumes
TR1	1 h	0.540	0.520	0.763 M	3.857 k	0.020	0.842 k	6
TR2	3 h	1.595	1.536	2.164 M	6.848 k	0.060	6.137 k	21
TR3	6 h	2.999	2.881	3.357 M	9.390 k	0.117	23.239 k	42
TR4	12 h	5.429	5.196	3.802 M	11.423 k	0.234	25.438 k	84
TR5	24 h	12.392	11.981	15.764 M	56.252 k	0.411	43.235 k	169
TR6	48 h	17.716	16.825	23.469 M	113.268 k	0.891	454.502 k	338

The raw radar source-data sizes in Table 14 represent the compressed or compact binary NEXRAD Level-II files stored on disk. These values are therefore substantially smaller than the amount of radar information represented after the files are decoded during processing and are useful measures of on-disk input size rather than direct measures of the number of observations processed or the memory required after decoding.

Across the six time ranges, ADS-B data account for approximately 95–97% of the reported raw on-disk source-data volume. This does not imply that ADS-B necessarily dominates computational cost or memory demand. NEXRAD Level-II data are stored in compact or compressed binary formats and expand during decoding, while runtime and memory demand also depend on the number, structure, and processing requirements of the observations contained in each dataset.

The computational evaluation investigates two processing strategies: feature selection (FS) and dataset partitioning (DP). Feature selection reduces the number of variables carried through the processing workflow by retaining only those required to produce the intended analytical model outputs. Dataset partitioning reduces the active processing workload by reading and reducing smaller temporal partitions before recombining the reduced data for operations that require the complete temporal context. DP therefore reduces the amount of source data that must be handled concurrently without changing the analytical interval represented by the final model.

Four scenarios compare feature selection and dataset partitioning, as summarized in Table 15. S1 disables both strategies and serves as the baseline; S2 isolates feature selection; S3 isolates dataset partitioning; and S4 combines both strategies.

Each scenario is evaluated in BUILD and LOAD execution modes. The cost of generating the required processed datasets and DGGS-based model outputs from source data is measured for the BUILD execution mode, and the cost of reusing previously generated processed source data and model outputs is measured for the LOAD execution mode.

The four processing scenarios are evaluated independently for each of the six historical workloads using Airspace model #2. This design isolates the computational effects of feature selection and dataset partitioning from changes in model geometry or DGGS resolution.

Table 15: The different scenarios used to evaluate computational efficiency in Experiment #1

Scenario	Description
S1	Baseline
S2	Feature selection only
S3	Dataset partitioning only
S4	Feature selection + dataset partitioning

The complete computational evaluation combines six historical workloads, four FS/DP processing scenarios, two execution modes, and four independent repetitions for every condition which produces the resulting summary experimental matrix below:

6 workloads × 4 processing scenarios × 2 execution modes × 4 repetitions = 192 valid executions

Computational performance is evaluated primarily using **total wall-clock time** and **process memory demand**. Wall-clock time measures the elapsed time required to complete a BUILD or LOAD execution. **Mean resident set size (RSS)** characterizes sustained process-memory demand, while **peak RSS** represents the maximum observed physical-memory demand. Stage-level wall-clock measurements are retained as diagnostic information for identifying computational bottlenecks. The file sizes of the saved ADS-B and weather model outputs are recorded to evaluate the effect of feature selection on storage and subsequent reuse. Output-completeness and preservation checks are evaluated separately so that apparent computational savings are not attributed to executions that fail to produce the required model outputs.

The definitive computational evaluation therefore consists of 192 valid executions. All measurements were obtained on a consumer-class laptop[1] using a fixed software environment and without dedicated high-performance-computing infrastructure. Visualization, risk-analysis, and clustering functions were disabled during the benchmark so that measured runtime and memory demand represented only model generation or model reuse. Median values across the four valid repetitions are used as the primary performance summaries.

Table 16 summarizes the variables retained when feature selection is enabled in scenarios S2 and S4. The baseline ADS-B dataset contains 17 source variables. Under FS, seven variables are retained during processing because velocity and last-contact time are required as temporary support variables, while five variables are written to the saved ADS-B model output: time, aircraft identifier, latitude, longitude, and geometric altitude. The baseline weather dataset contains ten variables. Under FS, six are retained during processing, including one temporary time variable required during storm processing, while five are written to the saved weather model output: longitude, latitude, height above sea level, reflectivity value, and sweep time. S1 and S3 retain the complete baseline variable sets.

[1] The computer used for all tests is a ASUS ROG Zephyrus G15 laptop with an AMD Ryzen 9 5900HS CPU, a NVIDIA GeForce RTX 3070 GPU, and 16 GB of RAM.

Table 16: Summary of the variables used for each scenario for testing the impact of FS

Dataset	Parameter	S1	S2	S3	S4
ADS-B dataset	time	✓	✓	✓	✓
	icao24	✓	✓	✓	✓
	lat	✓	✓	✓	✓
	lon	✓	✓	✓	✓
	velocity	✓		✓	
	heading	✓		✓	
	vertrate	✓		✓	
	callsign	✓		✓	
	onground	✓		✓	
	alert	✓		✓	
	spi	✓		✓	
	squawk	✓		✓	
	baroaltitude	✓		✓	
	geoaltitude	✓	✓	✓	✓
	lastposupdate	✓		✓	
	lastcontact	✓		✓	
	hour	✓		✓	
Weather radar dataset	longitude	✓	✓	✓	✓
	latitude	✓	✓	✓	✓
	heightASL	✓	✓	✓	✓
	heightRel	✓		✓	
	surfaceRan	✓		✓	
	radialAng	✓		✓	
	value	✓	✓	✓	✓
	elevAngle	✓		✓	
	sweepTime	✓	✓	✓	✓
	Sweep	✓		✓	

5.2 Airspace model implementation

This section describes how the two airspace model configurations defined in Section 5.1, Airspace Model #1 and #2, are generated from historical aerospace data. As shown in Figure 26, the implementation follows the first four methodology components introduced in Chapter 4: airspace entity and data source selection, data collection and preprocessing, the computational execution framework, and DGGS-based 4D airspace model construction. These steps produce the reusable airspace model outputs that are evaluated visually in Section 5.3 and computationally in Section 5.4.

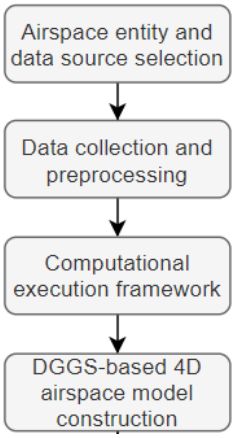


Figure 26: Overview of the airspace model methodology steps implemented in Experiment #1

5.2.1 Airspace entity and data source selection

Experiment #1 uses three data inputs: ADS-B aircraft trajectory data, NOAA NEXRAD weather radar data, and aircraft reference information where required for visualization and interpretation. These sources were selected to demonstrate how different types of historical aerospace data provided at different temporal rates and spatial patterns can be integrated into the same DGGS-based airspace model. ADS-B data represent moving aircraft as time-varying point entities. Weather radar data represent meteorological conditions as spatially distributed, time-varying observations, from which storm cell representations can be derived. Aircraft reference information provides metadata that support later interpretation of aircraft type and behaviour but is not required for the core DGGS cell assignment.

Table 17 summarizes the role of each data source in Experiment #1. The primary validation requirement is that both aircraft and weather data can be transformed into a common spatial and temporal representation. For this reason, the key processing outputs include not only processed datasets but also georeferenced and DGGS-indexed datasets that can be saved, loaded, visualized, and reused for downstream analysis.

Table 17: Role of each data source used in Experiment #1

Data source	Role in Experiment #1	Main processing outputs
ADS-B aircraft trajectory data	Represents historical aircraft motion	Filtered, smoothed, georeferenced aircraft trajectory points
NOAA NEXRAD radar data	Represents hazardous weather observations	Processed reflectivity data and storm cell outputs
Aircraft reference information, where used	Supports visualization and later interpretation	Aircraft metadata linked by aircraft identifier

5.2.2 Air traffic data collection and processing

Figure 27 illustrates the main steps of the air traffic data collection and processing including data import, coordinate transformation, filtering, smoothing, interpolating, and the removal of outliers.

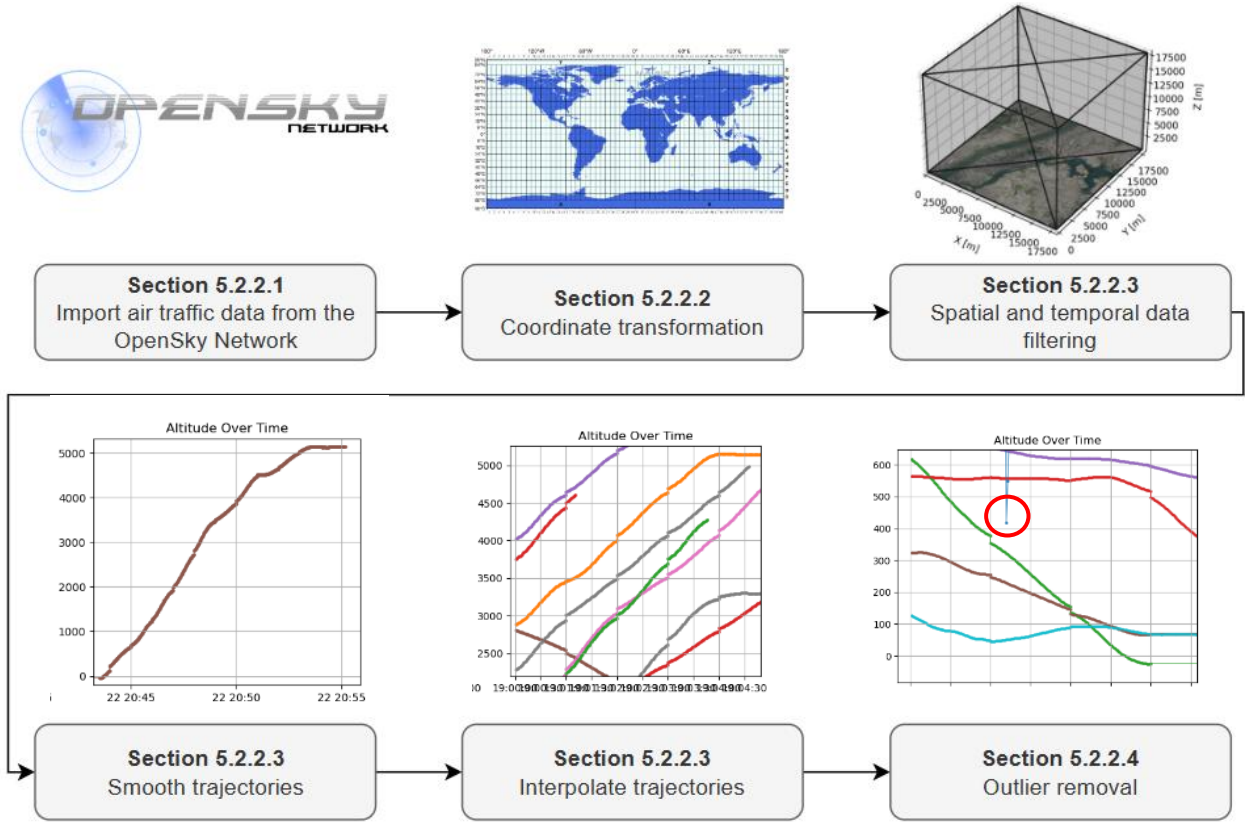


Figure 27: Air traffic data collection and processing

5.2.2.1 Import air traffic data from the OpenSky Network

Historical ADS-B air traffic data for a specified date and region of interest is imported from the OpenSky Network[2] [450]. Figure 28 shows the OpenSky Network ADS-B data coverage. The OpenSky Network is open-source and made readily available for researchers to use inside a Python environment via a series of software tools through a Trino SQL interface [451] backed by MinIO object storage.

The implemented Python retrieval workflow uses the pyOpenSky Trino interface [451] to query OpenSky historical state vectors. SQL queries restrict the selected time partitions and spatial bounds before returning rows to Python. Partition filtering is essential because the historical tables are organized by hour; accessing a specified interval through the corresponding hour partitions avoids an unrestricted scan of the complete historical collection [450], [451].

[2] The OpenSky Network only surveils 1090 MHz SSR Mode S downlink channel ADS-B traffic and not the 978 UAT. While 100% surveillance of all aircraft is not feasible for the OpenSky Network, there is a known coverage gap in the network for aircraft equipped with 978 UAT-type ADS-B.

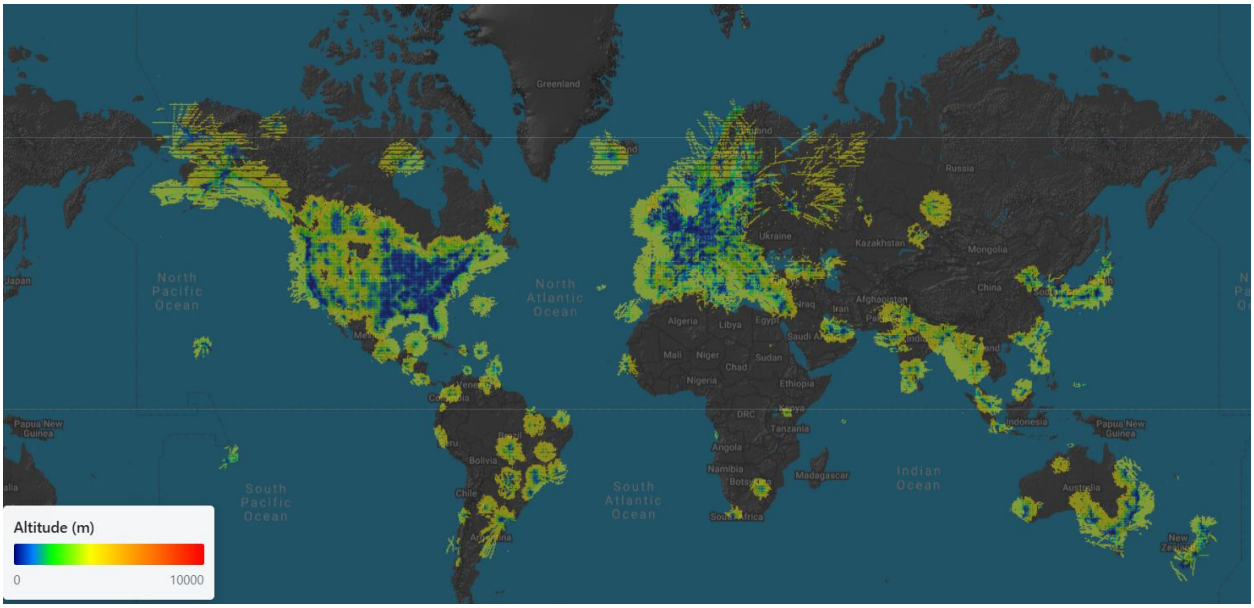


Figure 28: OpenSky Network ADS-B data coverage [452]

Figure 29 illustrates the historical ADS-B data acquisition and storage workflow implemented in Python. The process begins with the configuration parameters defining the selected geographical region and UTC time interval. The Python pipeline uses the pyOpenSky Trino client to submit SQL queries to the OpenSky Network's Trino query engine, which accesses the historical ADS-B data stored in its MinIO-based infrastructure. Queries retrieve selected aircraft state-vector attributes from the state_vectors_data4 table, using hourly data partitions and temporal and geographical filters to efficiently access the required observations. The retrieved state vectors are then returned to Python as Pandas dataframes for local processing.

The retrieved observations are organized and saved locally as minute-level SQLite database files containing the original aircraft state-vector records. These observations are subsequently processed through geographical and temporal filtering, duplicate removal, trajectory segmentation, and smoothing and interpolation before being saved as processed minute-level CSV files. Both datasets are organized using a hierarchical directory structure based on their timestamps. The Python pipeline also identifies previously saved files corresponding to the requested time interval and reuses the available observations during subsequent executions. This approach facilitates the retrieval, processing, persistent storage, and reuse of historical ADS-B data for airspace model generation.

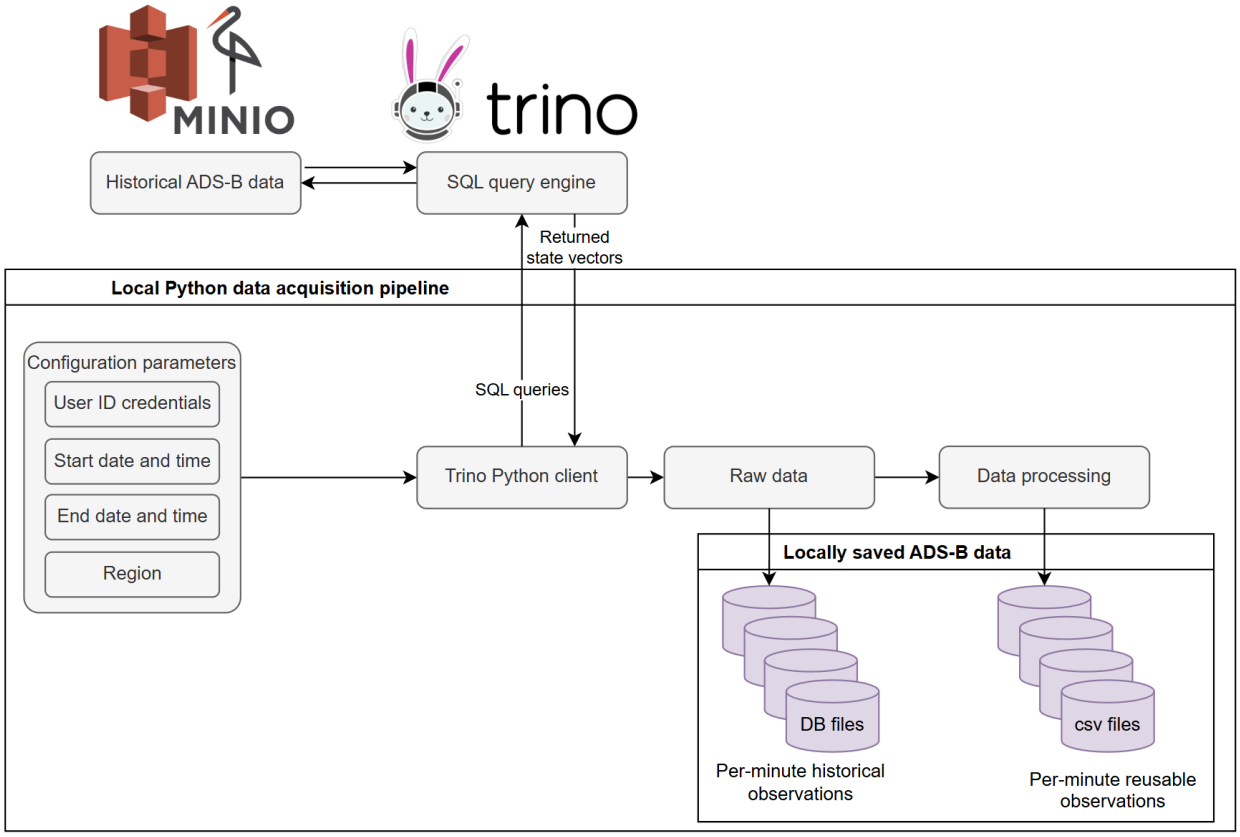


Figure 29: The process for accessing data using Trino

Table 18 lists the parameters available in the OpenSky Network ADS-B database, their data types, and their descriptions. From these available parameters, only the lon (longitude), lat (latitude), geoaltitude (geometric altitude), icao24 (international unique aircraft ICAO identifier), and time parameters are kept and are relevant to the current research.

Table 18: List of available parameters in the OpenSky Network's dataset [450]

Parameter	Type	Description
time	Unix timestamp	The time at which the state vector was valid
icao24	string	The unique 24-bit transponder ID assigned to the aircraft.
lat	double	The last known latitude (WGS84 format).
lon	double	The last known longitude (WGS84 format).
velocity	double	Speed over ground in meters per second.
heading	double	The direction of movement in degrees from geographic north.
vertrate	double	Vertical speed in meters per second (positive = ascending, negative = descending).
callsign	string	The flight identifier broadcast by the aircraft.
onground	boolean	Indicates if the aircraft is on the ground (true) or airborne (false).
alert	boolean	Special ATC indicator
spi	boolean	Special ATC indicator
squawk	string	The 4-digit octal transponder code assigned by ATC.
baroaltitude	double	Barometric altitude measured by the aircraft.
geoaltitude	double	Geometric altitude determined by GNSS (GPS).
lastposupdate	double	The timestamp of the last recorded position update.
lastcontact	double	The last time OpenSky received a signal from the aircraft.
hour	int	The start of the hour this data belongs to.

5.2.2.2 Coordinate transformation

The longitude, latitude, and geometric altitude parameters are transformed to the UTM coordinate system. The UTM system divides the Earth into 60 longitudinal zones, each 6° wide. Each zone has a central meridian where distortion is minimized. UTM coordinates are expressed as Easting and Northing, where Eastings correspond to the x-coordinate, relative to the central meridian and Northings correspond to the y-coordinate, relative to the equator. Altitude remains unchanged in this conversion and can be included in UTM as-is. Snyder [449] developed the following process for converting the longitude and latitude parameters to UTM:

Step#1: Determining the UTM zone

The UTM zone is calculated from the longitude (λ) using Equation (1):

$$zone = \left\lfloor \frac{\lambda + 180}{6} \right\rfloor + 1 \tag{1}$$

where λ is in decimal degrees and the $\lfloor \cdot \rfloor$ operator indicates the floor function.

Step#2: Converting degrees to radians

Equations (2) and (3) are used to convert geographic coordinates from degrees to radians.

$$\phi = \text{latitude in radians} = \text{latitude in degrees} \times \frac{\pi}{180} \tag{2}$$

$$\lambda = \text{longitude in radians} = \text{longitude in degrees} \times \frac{\pi}{180} \tag{3}$$

Step#3: Calculating the central meridian of the UTM zone

The central meridian (λ_0) for a given UTM zone is given by Equation (4):

$$\lambda = (zone - 1) \times 6 - 180 + 3 \; (in \; degrees) \tag{4}$$

The central meridian (λ_0) is converted to radians using Equation (5):

$$\lambda_0 = \lambda_0 \times \frac{\pi}{180} \tag{5}$$

Step#4: Computing UTM coordinates

For the WGS84 ellipsoid [453], which is a mathematical model of Earth's shape and size used as the reference frame for GPS and many geospatial systems, the semi-major axis parameter a is 6378137 m, the flattening parameter f is 1/298.257223563, and the eccentricity squared parameter e^2 is $2f - f^2 = 0.006694379990141316 5$. The $\mathrm{N}, T, C, A, and\ M$ parameters for the UTM projection are calculated using Equations (6) through (10):

$$\mathrm{N} = \frac{a}{\sqrt{1 - e^2 \sin^2(\phi)}} \tag{6}$$

$$\mathrm{T} = \tan^2(\phi) \tag{7}$$

$$C = \frac{e^2}{1-e^2}\cos^2(\phi) \tag{8}$$

$$A = (\lambda - \lambda_0)\cos(\phi) \tag{9}$$

$$M = a\left[\left(1 - \frac{e^2}{4} - \frac{3e^4}{64} - \frac{5e^6}{256}\right)\phi - \left(\frac{3e^2}{8} + \frac{3e^4}{32} + \frac{45e^6}{1024}\right)\sin(2\phi) + \left(\frac{15e^4}{256} + \frac{45e^6}{1024}\right)\sin(4\phi) - \frac{35e^6}{3072}\sin(6\phi)\right] \tag{10}$$

The scale factor at the central meridian is k_0 is 0.9996. The Easting and Northing X and Y coordinates are calculated using Equations (11) and (12):

$$X = k_0 N\left(A + \frac{(1-T+C)A^3}{6} + \frac{(5-18T+T^2+72C-58e^2)A^5}{120}\right) + 500{,}000 \tag{11}$$

$$Y = k_0 N\left[\left(M + Ntan(\phi)\left(\frac{A^2}{2} + \frac{(5-T+9C+4C^2)A^4}{24} + \frac{(61-58T+T^2+600C-330ee^2)A^6}{720}\right)\right)\right] \tag{12}$$

For the southern hemisphere, 10,000,000 m is added to Y to ensure positive values.

5.2.2.3 Data filtering, smoothing, and interpolation

A filter is applied to remove datapoints outside of the selected airspace regions and desired start and end times for the airspace models listed in Table 13.

Another filter is applied to remove all aircraft operating on the ground. The *onground* parameter values (see Table 18) are used to detect aircraft on the ground and remove them from the processed dataset.

The time intervals between successive datapoints are at irregular intervals but are on average less than 1 second. Each datapoint is grouped in individual trajectories using the unique aircraft ICAO identifier and resampled for every 1 second time interval to provide enough granularity for all results presented using the air traffic data.

For each trajectory, a Savitzky-Golay (SG) filter is applied to smooth out any noise across the 1 second time intervals. The SG filter works by performing a polynomial least-squares fit over a sliding window of data points [454]. The resulting polynomial coefficients are used to smooth the central point of the window [454]. Mathematically, for a signal $y[n]$, the filtered value $y'[n]$ at the index n is given by Equation (13):

$$y'[n] = \frac{\sum_{k=-m}^{m} c_k y[n+k]}{\sum_{k=-m}^{m} c_k} \tag{13}$$

Where c_k are the convolution coefficients, m is the half-window size (the total window size is $W = 2m + 1$), and $y[n+k]$ are the original data points within the window. The convolution coefficients c_k are computed based on the order p of the polynomial and the window size W. The coefficients are derived to

minimize the least-squares error of fitting the polynomial to the data points. The coefficients c_k are derived from solving the least-squares polynomial fitting problem using Equation (14):

$$c_k = \sum_{j=0}^{p} \frac{\binom{p}{j}(-k)^j}{\sum_{i=-m}^{m}\binom{p}{j}(-i)^j} \tag{14}$$

The window size W determines the trade-off between smoothing and preserving signal features. For the ADS-B dataset, processed to every 1 second, a window size of 5 is used to capture ±2 seconds around each point.

Higher values of the polynomial order p will capture more details and variations in the signal because the polynomial can adapt to more complex features but can be subject to overfitting, while lower values will produce smoother results by fitting simpler polynomials (linear or quadratic trends) but can be subject to over-smoothing. A value of 3 is chosen to preserve general trends in the dataset while reducing noise.

5.2.2.4 Outlier removal

Trajectory quality control is applied before the processed ADS-B data are mapped into the airspace model. Published aircraft-performance limits [455], [456] were used during development as plausibility references for identifying unrealistic aircraft motion, but the implemented preprocessing does not apply a single fixed acceleration cutoff criterion to classify observations as outliers. Instead, the preprocessing identifies large temporal or spatial discontinuities, separates discontinuous observations into different trajectory segments, and smooths valid trajectory coordinates where sufficient observations are available.

The purpose of this step is to prevent clearly inconsistent observations or trajectory discontinuities from propagating into interpolation, DGGS assignment, and subsequent analyses.

5.2.2.5 Data storage for the processed dataset

The historical ADS-B observations retrieved from the OpenSky Network are saved locally using a time-partitioned file structure organized by year, month, day, and hour. Within this structure, the retrieved observations are stored in individual minute-level SQLite database files containing the original aircraft state-vector records. This organization allows the processing pipeline to identify and load the files corresponding to a selected time interval and reuse previously retrieved observations. The selected records are subsequently processed using Pandas, and the resulting datasets are saved as minute-level CSV files. The processed datasets can then be loaded and reused during subsequent model generation and analysis. The corresponding local file organization is illustrated in Figure 30.

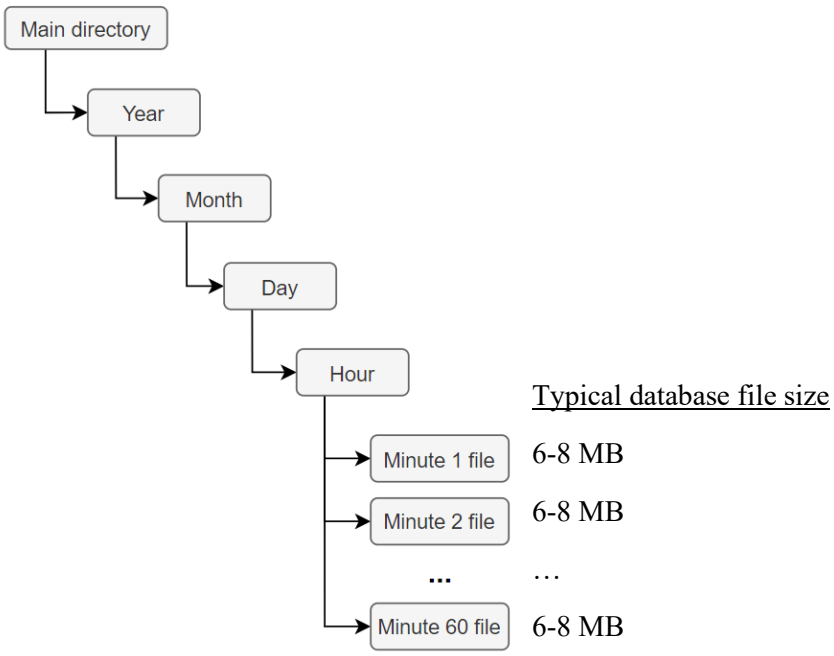


Figure 30: Local dataset storage structure

The local SQLite database files provide persistent storage of historical ADS-B observations, while the processed CSV files preserve the results of the data preparation stage. Both are organized into minute-level files within a hierarchical directory structure based on year, month, day, and hour. During subsequent executions, the required files are identified according to the selected temporal interval, and their contents are loaded into Pandas dataframes. SQLite supports selective retrieval of the required data attributes, while the time-partitioned file structure allows the processing pipeline to access the relevant observations directly. This organization supports efficient temporal data retrieval, persistent storage, and reuse of previously generated datasets for different model configurations and analytical applications. Figure 31 shows an example of a locally saved SQLite database file viewed using database software.

DB Browser for SQLite - G:\My Drive\PhD\Thesis\adsb_1776710160_figure34.db

File Edit View Tools Help

New Database | Open Database | Write Changes | Revert Changes | Undo | Open Project | Save Project | Attach Datab

Database Structure | Browse Data | Edit Pragmas | Execute SQL

	time	icao24	lon	lat	geoaltitude	X_UTM	Y_UTM	UTM_zone
1	1776710160	04c11a	-73.8639554110441	40.6277207196769	548.64	596079.612	4498052.137	18N
2	1776710160	4c0221	-73.4667358398438	40.9490661621094	4884.42	629051.282	4534234.777	18N
3	1776710160	a00c7a	-74.1399536132813	40.4949645996094	1645.92	572880.93	4483051.026	18N
4	1776710160	a02ed4	-73.9119262695313	40.373291015625	1607.82	592370.826	4469758.39	18N
5	1776710160	a05f95	-73.588735407049	41.0450149794756	3810.0	618610.401	4544713.486	18N
6	1776710160	a067f6	-74.0578391335227	40.9185325493247	563.88	579335.17	4530140.725	18N
7	1776710160	a08714	-73.8613891601563	40.6002960205078	868.68	596336.041	4495010.653	18N
8	1776710160	a12f70	-74.2006225585938	40.6355895996094	266.7	567598.109	4498612.443	18N
9	1776710160	a13c73	-73.4938354492188	40.4293212890625	1013.46	627759.341	4476498.593	18N
10	1776710160	a145b7	-73.9516587690874	40.594994496491	3779.52	588705.343	4494327.288	18N
11	1776710160	a16ebc	-74.2072143554688	40.8003845214844	2194.56	566875.551	4516900.565	18N
12	1776710160	a18380	-73.8564453125	40.4995880126953	3307.08	596899.412	4483837.056	18N
13	1776710160	a1aaae	-73.7088359485973	40.5246540651483	2827.02	609366.708	4486792.122	18N
14	1776710160	a1caf4	-73.4097290039063	40.6988983154297	266.7	634353.63	4506548.789	18N
15	1776710160	a1d45a	-73.9920460094105	40.9013547735699	2423.16	584897.367	4528295.576	18N
16	1776710160	a1d49e	-73.9949174360796	40.7048112255032	251.46	584905.636	4506474.846	18N
17	1776710160	a1e902	-74.178466796875	40.6709289550781	45.72	569435.05	4502552.532	18N
18	1776710160	a2008f	-73.4150765158913	40.8177004022113	441.96	633663.589	4519728.982	18N
19	1776710160	a2380f	-74.261962890625	40.72265625	548.64	562329.761	4508231.924	18N
20	1776710160	a24a8d	-74.106689453125	40.8000183105469	274.32	575355.854	4516941.449	18N
21	1776710160	a2a536	-74.0286254882813	40.714307882018	205.74	582046.405	4507497.001	18N
22	1776710160	a2bf05	-74.3153936212713	40.8820821471133	320.04	557679.134	4525892.74	18N
23	1776710160	a3245d	-74.0725083784624	40.8417213569253	5173.98	578190.268	4521600.767	18N
24	1776710160	a39ada	-73.650634765625	40.5942993164063	571.5	614178.308	4494596.978	18N

Figure 31: Example of a processed database file

5.2.3 Weather radar data collection and processing

Figure 32 summarizes the weather radar data collection and processing workflow used in Experiment #1. Historical NOAA NEXRAD Level-II radar volumes are imported from AWS and decoded into radar data. These radar points are then geolocated and transformed into the airspace model. Reflectivity, spatial, vertical, and temporal filtering are subsequently applied to retain only the radar points relevant to the selected airspace-model domain and analysis objective. The retained radar points are then grouped into storm cell structures; points not retained by the clustering stage are removed as outliers/noise; and the identified storm cells are summarized using time-ordered centroid positions to generate simplified storm cell trajectories for use in the airspace model.

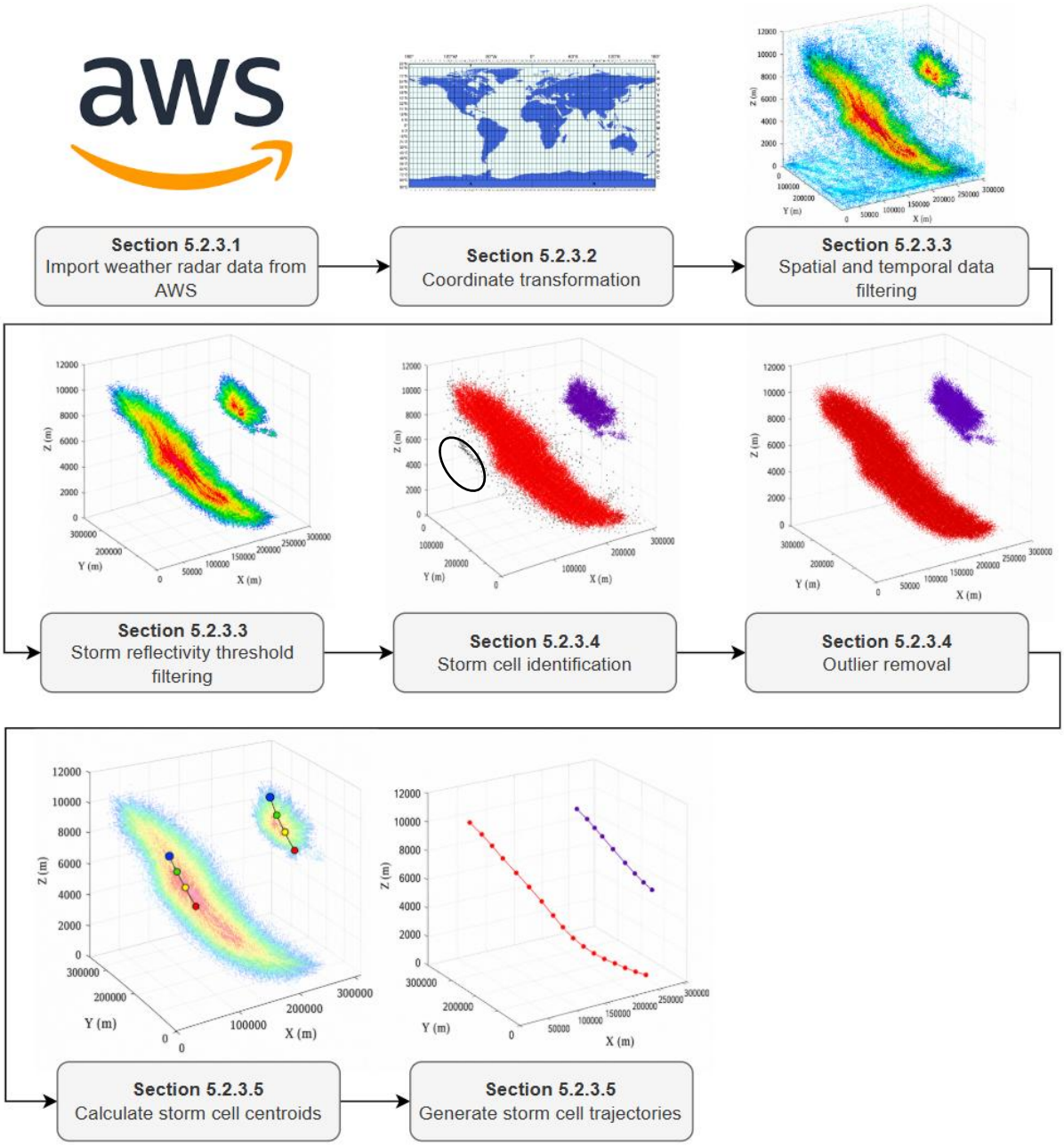


Figure 32: Weather data collection and processing

5.2.3.1 Import weather radar data from AWS

Historical weather radar data for a specified date and region of interest is imported from the National Oceanic and Atmospheric Administration (NOAA). The National Centers for Environmental Information (NCEI) [457] is a U.S. government agency that manages one of the world's largest archives of atmospheric, coastal, geophysical, and oceanic data, including NOAA's NEXRAD II weather radar. Weather radar data is archived based on the time and the weather station where the data was collected. This makes it possible to extract only the data that is within a specific geographical boundary for a specific time.

Figure 33 shows the NOAA NEXRAD weather radar data coverage in green shading. The figure shows that most of the continental United-States is covered by NEXRAD radar, with some notable gaps in the western parts of the country, particularly over the Rocky Mountains (Montana, Wyoming, Colorado, New Mexico) and Sierra Nevada (eastern California and western Nevada), where terrain elevation obstructs radar line-of-sight.

Figure 34 shows the KOKX weather radar station location (yellow square) and coverage (green region), which is the radar station that is used to gather radar data for the two airspace model configurations used in Experiment #1.

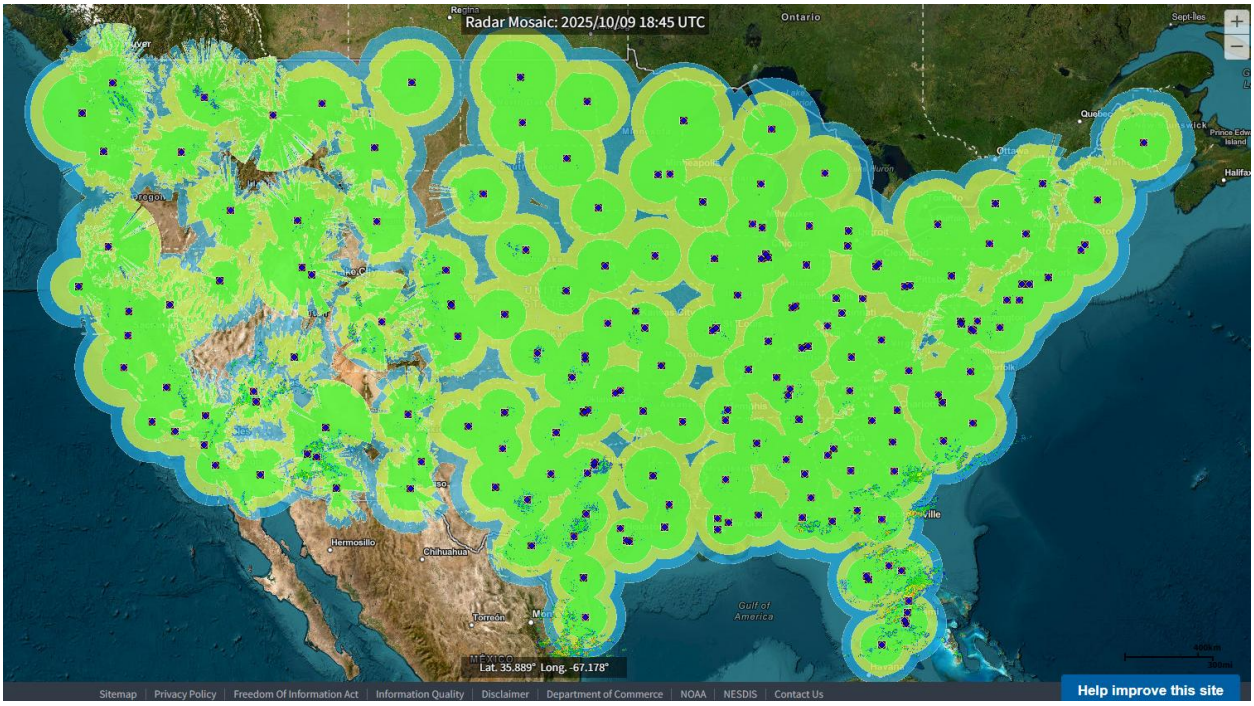


Figure 33: NOAA NEXRAD weather radar data coverage [458]

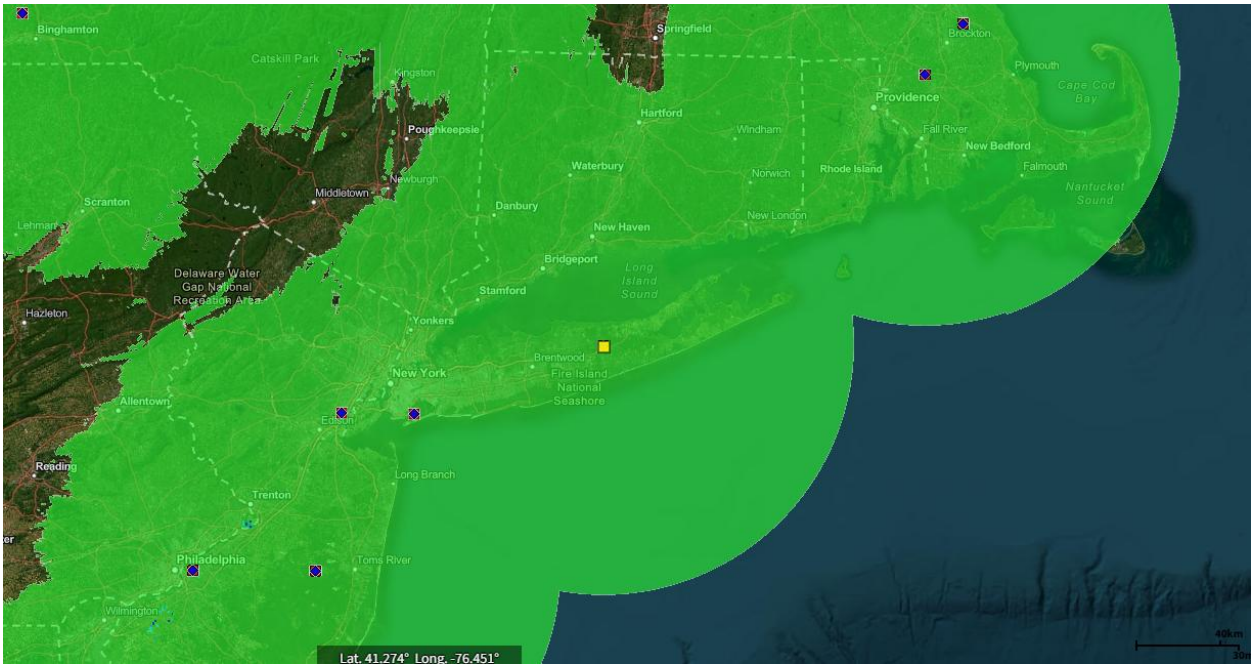


Figure 34: KOKX weather radar station location (yellow square) and coverage (green region) [458]

The NCEI makes its database accessible to the public through multiple platforms, and for Experiment #1, radar data was extracted from Amazon Web Services (AWS) servers from the NCEI Amazon S3 bucket [459]. Amazon Web Services is a subsidiary of the company Amazon that provides businesses, governments, and individuals with on-demand cloud computing platforms and Application Programming Interfaces (APIs), programs that allow one piece of software to communicate with another. AWS offers a data storage service named Amazon Simple Storage Service (Amazon S3) which uses an object storage architecture that treats data as an object and can organize many objects amongst different "buckets" [460]. The advantage of selecting this data source is the potential to automate the data extraction process.

Figure 35 shows an overview of the process of importing NOAA weather radar data through Amazon S3 servers. A Python program was developed which is used to communicate using an AWS API to the Amazon S3 buckets containing the relevant weather radar data for the desired model configuration (i.e., airspace region and date and time range). The Python program stores the user ID credentials (username and password) used to make requests to the Amazon S3 servers. The Amazon S3 bucket name containing the data is called "noaa-nexrad-level2" and the bucket region used in this case corresponds to the weather station identification name. Because the radar data needs to be gathered based on a provided weather station, a check must first be performed to determine which weather stations are located inside of the region of interest for the use case of the airspace model. If multiple weather stations are located inside of the region of interest, additional processing of the data must be done to perform data fusion between the different stations. For the scope of this thesis, only regions containing one single weather station are used. The data collected from the Amazon S3 bucket comes in a NEXRAD II encoded format that must first be decoded to made useful to an end user. The NEXRAD Level-II archive files are accessed and downloaded from the AWS S3 archive using the Boto3 API [461]. The downloaded radar files are then decoded and read in Python using the Python ARM Radar Toolkit (Py-ART) [462], after which the required radar fields are converted into analysis-ready outputs for filtering, storm cell extraction, DGGS mapping, and local storage using the .db file format.

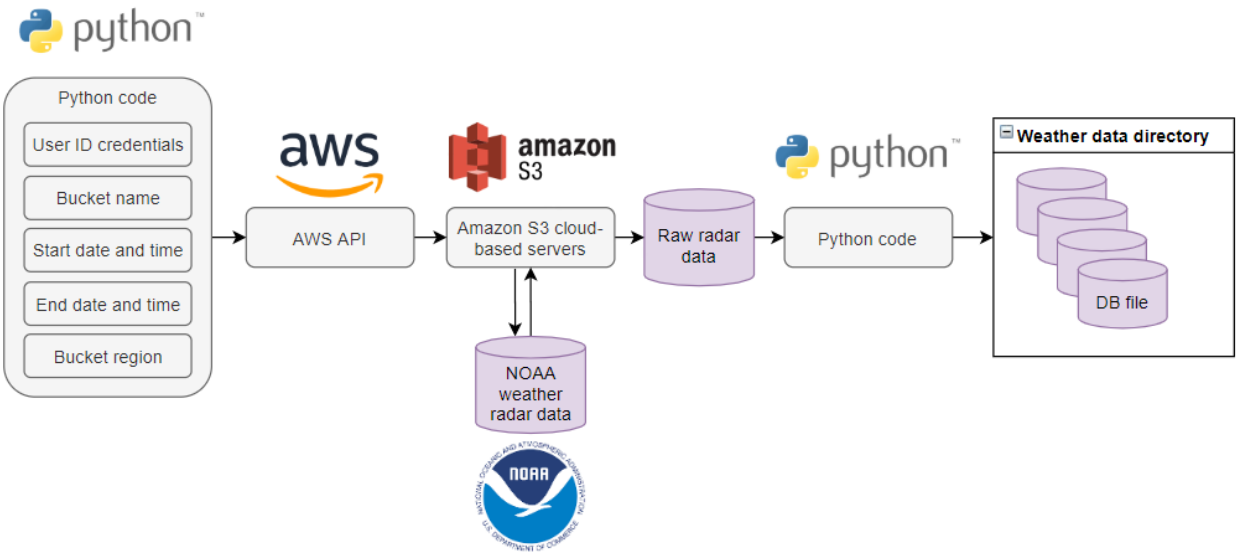


Figure 35: The process for importing weather data from Amazon S3 servers

Table 19 lists the parameters contained in the NOAA data, descriptions of their units, their data types, and example values for each. From these available parameters, only the longitude, latitude, heightASL (height above sea level), value (the base reflectivity value), and sweepTime (the time of the radar sweep) parameters are retained and are relevant to this thesis.

Table 19: List of available parameters in the NOAA dataset [463]

Parameter	Type	Description
longitude	double	Longitude of the base reflectivity data point in angular degrees
latitude	double	Latitude of the base reflectivity data point in angular degrees
heightASL	double	Height above sea level in meters
heightRel	double	Height relative to radar in meters
surfaceRan	double	Range of radar site in meters
radialAng	double	Azimuth angle in the radar's polar coordinate system, where 0 is north in angular degrees
value	double	Base reflectivity value in dBZ
elevAngle	double	The tilt angle in angular degrees between the horizontal plane and the line of sight of the current sweep
sweepTime	string	Time of the sweep in UTC using ISO 8601 standard format (YYYY-MM-DDT Hh:mm:ssZ)
Sweep	int	Radar sweep ranging from 0 to 360 degrees

5.2.3.2 Coordinate transformation

The longitude, latitude, and heightASL parameters are transformed to the UTM coordinate system using the same process used for the air traffic data as described in Section 5.2.2.2.

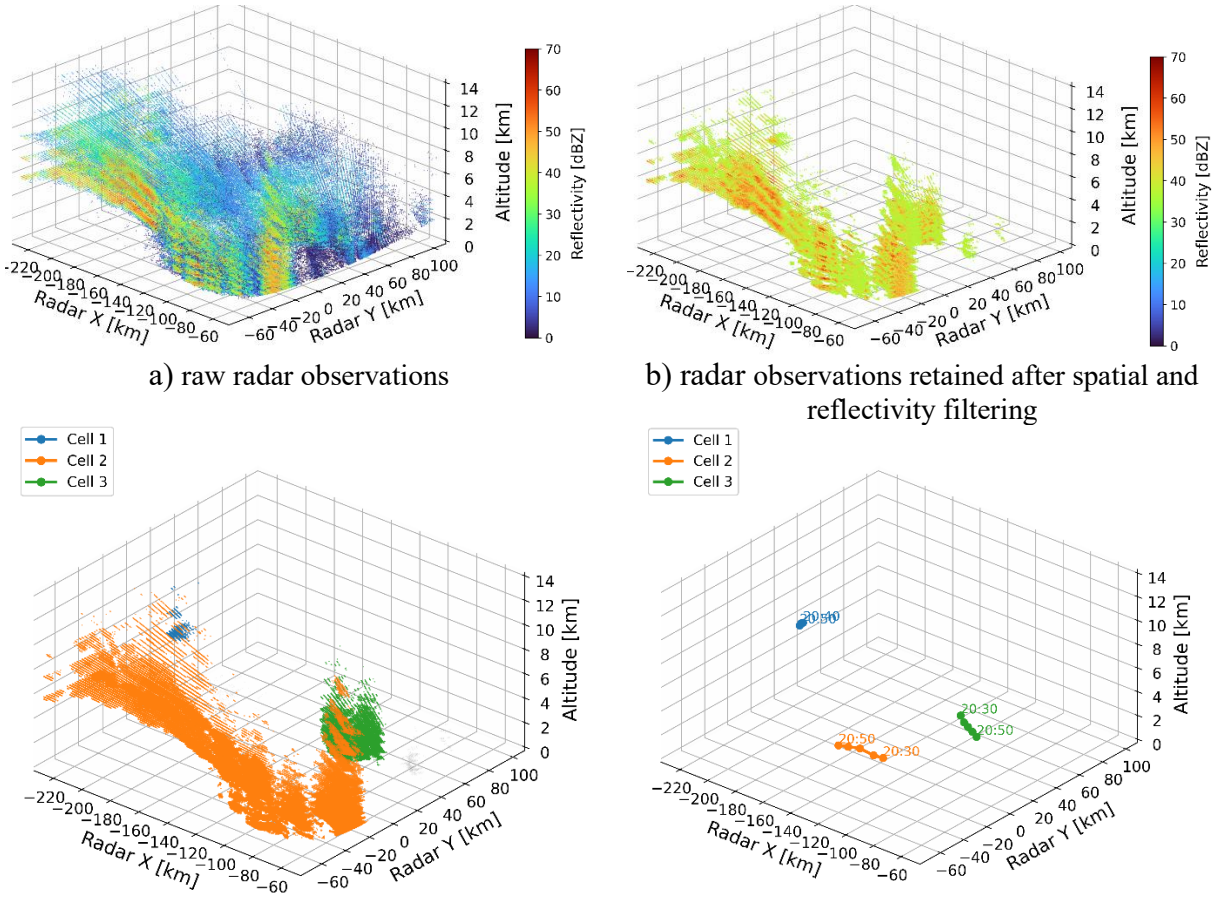
5.2.3.3 Spatial, temporal, and storm reflectivity filtering

After import and coordinate processing, radar observations are filtered to retain only the data relevant to the selected airspace-model domain and analysis interval. Three forms of filtering are applied. First, a

reflectivity threshold is used to retain only observations associated with meteorologically significant weather points. Second, spatial and altitude filtering remove observations outside the horizontal and vertical bounds of the selected airspace model. Third, temporal filtering retains only radar observations within the selected analysis interval. These operations reduce the volume of radar data carried into the subsequent storm cell identification stage while preserving the observations relevant to the weather structures of interest.

5.2.3.4 Storm cell identification and outlier removal

The filtered radar points are grouped into storm cell structures using a density-based clustering procedure. The purpose is to identify spatially and temporally related groups of reflectivity observations that correspond to individual storm cells within the selected airspace-model domain. Observations that are not retained as part of the identified storm cell structures are treated as outliers/noise and removed from the subsequent storm cell summarization step. Figure 36 illustrates this transformation from filtered radar observations to grouped storm cell structures. Figure 36a shows the raw radar observations, and Figure 36b shows the observations retained after spatial, temporal, and reflectivity filtering.



a) raw radar observations

b) radar observations retained after spatial and reflectivity filtering

c) storm cell assignments during DBSCAN clustering

d) final storm cell centroids

Figure 36: 3D visuals of the sequential processing steps of weather data from raw reflectivity values to storm cell centroids

5.2.3.5 Storm cell centroid and trajectory generation

For each identified storm cell, representative three-dimensional centroid positions are calculated at successive time steps to produce a simplified storm cell representation. The centroid positions summarize the spatial evolution of each storm cell while substantially reducing the complexity of the full radar point cloud. Ordering these centroid positions by time yields a simplified storm cell trajectory that can be stored, visualized, and later integrated into the DGGS-based airspace model. Figure 36c shows the storm cell centroids derived from the identified cells after DBSCAN clustering, while Figure 36d shows the resulting time-ordered storm cell trajectories.

5.2.4 Computational execution framework

The computational execution framework controls how the airspace model is generated, saved, and reused. It is implemented before DGGS-based model construction and determines whether the required datasets and model artifacts are generated from source data or loaded from previously saved outputs.

In BUILD mode, the required data products and model outputs are regenerated from source or intermediate data. In LOAD mode, previously generated outputs matching the selected model configuration are loaded from local storage. BUILD therefore measures the cost of constructing the model, whereas LOAD measures the cost of reusing an existing persisted model.

Dataset partitioning is implemented as a time-based ingestion and memory-management strategy for both ADS-B and weather data. Source data are read and reduced within smaller temporal partitions so that the complete historical source workload does not need to be retained concurrently during these stages. Reduced partition outputs are subsequently recombined before processing operations that require global temporal context. For ADS-B data, trajectory segmentation, smoothing, and final preprocessing are therefore applied after the staged partitions are recombined. For weather data, radar partitions are processed individually before the reduced radar outputs are recombined and storm cell information is derived from the complete analytical interval. This design reduces the amount of source data handled concurrently while avoiding artificial trajectory or storm boundaries introduced solely by the partitioning scheme.

Partitioning introduces additional processing operations, including partition management, repeated source access at selected boundaries, temporary staging, and recombination. DP is therefore expected to reduce memory demand rather than necessarily reduce total execution time. The computational effects of this trade-off are evaluated in Section 5.4.

The computational execution framework produces the same required analytical model outputs under the four FS/DP scenarios defined in Section 5.1. The performance of these strategies is evaluated in Section 5.4 using total wall-clock time, mean and peak resident set size (RSS), and saved model-output size. Model-reuse performance is evaluated through the wall-clock time required to load previously generated outputs.

5.2.5 DGGS-based airspace model construction

In the final step, the DGGS-based 4D airspace model is constructed, and the processed airspace datasets are mapped into it. The DGGS framework provides the common spatial and temporal structure used to represent aircraft trajectory points, radar observations, and storm cell objects.

The DGGS grid is generated from the model parameters defined in Section 5.1. These parameters include the 3D model boundary, DGGS cell size, and temporal interval. After the DGGS grid is generated, processed ADS-B, radar, and storm cell data are assigned to DGGS cells using their UTM coordinates and altitude. The corresponding time value is retained so that the resulting datasets represent both spatial

position and temporal evolution. This creates a common indexing structure in which aircraft and weather observations can be compared, aggregated, visualized, and reused for downstream analyses.

Each processed observation is assigned to a DGGS cell using its three-dimensional coordinates and the corresponding grid resolution. The cell-mapping procedure uses integer spatial indices along the three axes, with lookup dictionaries organized by resolution to efficiently identify candidate cells and verify their spatial bounds. Each mapped observation is then associated with the corresponding DGGS centroid identifier and Morton code. Morton codes are calculated by interleaving the integer indices of the three spatial axes, producing a one-dimensional ordering that preserves spatial locality among neighbouring cells (see Appendix A for more details). This spatial indexing mechanism provides an organized relationship between the DGGS structure and the heterogeneous observations associated with its cells.

The same indexing and data organization principles support subsequent location-based calculations. In the risk models, the DGGS centroids are sorted according to their Morton codes and divided into spatially ordered computational batches. Aircraft observations are first organized by aircraft identifier and timestamp, allowing consecutive positions to be associated, and are subsequently grouped by time for efficient retrieval during each calculation step. At each time step, the relevant aircraft observations are retrieved and used to calculate their convergence toward the DGGS centroids, with the collision risk calculations distributed across multiple processing workers. Weather risk calculations follow a similar organization, using storm observations grouped by identifier and time and processing the Morton-ordered centroids in sequential batches. The resulting analytical values are saved as new csv files with their corresponding centroid identifiers and timestamps, enabling their subsequent retrieval, comparison, and reuse across different analytical applications.

The outputs of this step include the DGGS grid, the processed and georeferenced aircraft dataset, the processed radar dataset, the storm cell dataset, the DGGS-indexed versions of these datasets, and the saved model artifacts used by LOAD mode. Together, these outputs constitute the reusable DGGS-based airspace models, which are evaluated in the remainder of this chapter and subsequently used in the experiments presented in Chapters 6 and 7.

5.3 Airspace model development results and visualization

This section presents the airspace model-development outputs for the two airspace model configurations defined in Section 5.1 (Airspace Model #1 and #2). The purpose is to verify that the implementation described in Section 5.2 produces a three-dimensional, time-indexed DGGS representation from historical ADS-B aircraft trajectory data and NOAA NEXRAD weather radar data. The figures are therefore used as model-verification outputs rather than as independent analytical results. They demonstrate that the source datasets were processed, georeferenced, mapped into the DGGS structure, and represented in both horizontal and three-dimensional forms.

Airspace model #1 evaluates the methodology within a compact, high-resolution terminal environment centred on LGA, whereas Airspace model #2 evaluates the same methodology over a larger regional domain using mixed DGGS resolution. Together, the two configurations provide evidence that the same modelling workflow can be applied at different spatial extents and resolution strategies.

5.3.1 Airspace model #1 results

5.3.1.1 Airspace model #1 air traffic - 2D scatter plot

Figure 37 shows scatter points of ADS-B trajectories over time for the 5 km by 5 km area around the LGA airport, overlaid on a Mapbox basemap which highlights runways and taxiways in green. The figure

presents 5 hours of cumulative ADS-B data for a high traffic scenario from 2022-11-22 19:00 to 2022-11-22 24:00 local time. Each scatter point is coloured using a continuous colour scale based on each aircraft's geometric altitude in meters, where dark red points show lower altitudes and dark blue points show higher altitudes. The figure provides an overview of where the air traffic is concentrated and what major traffic patterns are found in the data.

High density of distinct landing and takeoff patterns: The cumulative trajectories show concentrated arrival traffic aligned with Runway 22 and departure traffic associated with Runways 13 and 31. Because the figure aggregates five hours of traffic, operations associated with both Runways 13 and 31 appear in the same visualization even though these opposing runway directions are not used simultaneously for the same traffic flow. Arrival and departure patterns are distinguished using trajectory direction and vertical motion. Black arrows identify the principal departure directions, while blue arrows identify the principal arrival direction.

Multiple air vehicle types in the terminal area: The figure also shows helicopter flyovers crossing the LGA airspace from East to West or West to East (see green arrow) at approximately 500 m altitudes along a known helicopter air route. Another air traffic pattern observed in Figure 37 is highlighted by the two yellow arrows. This pattern shows aircraft flying in a linear fashion at higher altitudes in the 2000 – 4000 m altitude range along or near known air routes.

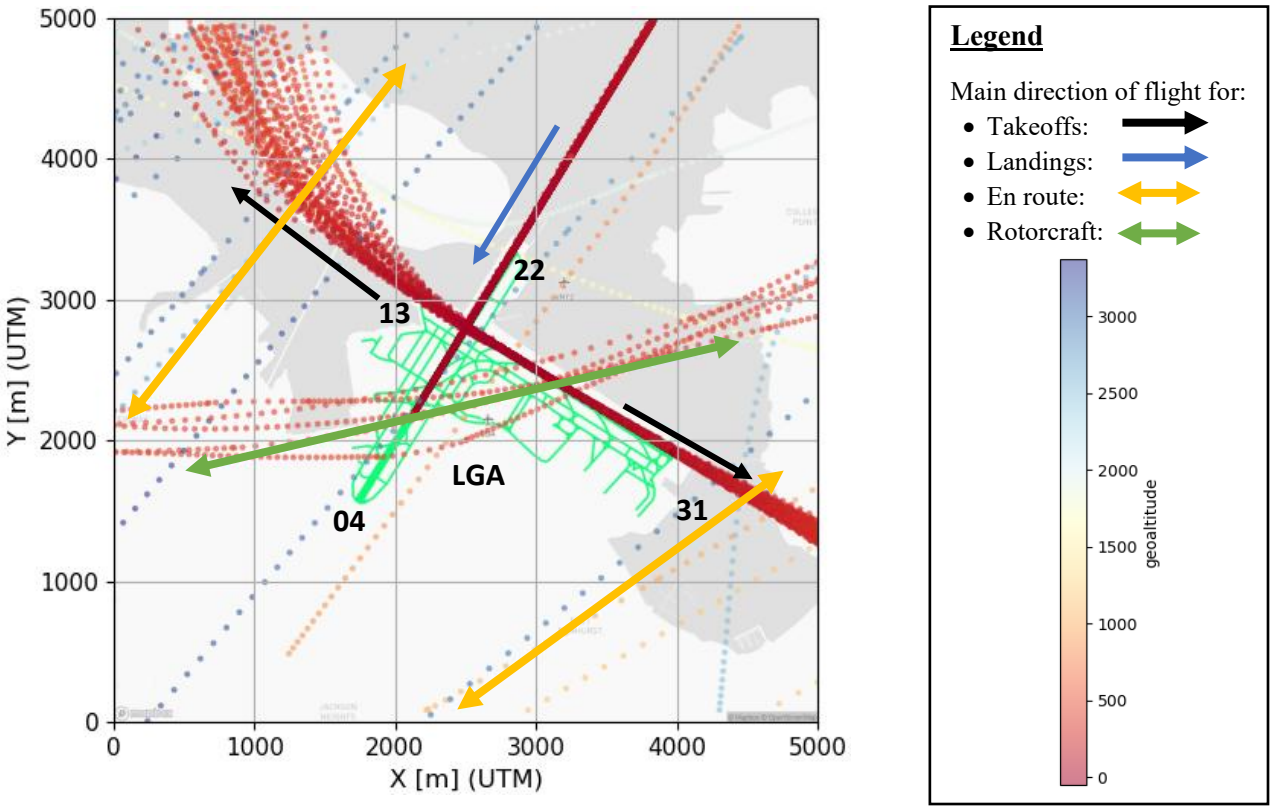


Figure 37: 2D map of air traffic data for Airspace model #1

5.3.1.2 Airspace model #1 air traffic - 3D scatter plot

Figure 38 shows the 3D version of Figure 37, and can be used to make more complex observations about the data in the model.

Mixed aircraft type categories and operational patterns: Each scatter point is identified as a small, medium, large fixed-wing aircraft or rotorcraft using the ICAO aircraft type identifier from the c OpenSky Network Aircraft Database [59] and the FAA's Aircraft Registration Database [464]. This identification gives an idea of the types of air vehicles that operate in the airspace region of Airspace model #1 and their spatial distribution. The figure shows that the different air vehicle types form visually identifiable operational patterns across different altitude ranges. Three main altitude groups are observed: very low (~ 0 to 600 m), mid-level (~500 to 1000 m), and high-altitude traffic (~2000 to 5000 m).

Low-level patterns: The lowest altitude band is primarily composed of medium and large fixed wing aircraft taking off and landing and is illustrated by the orange and blue scatter points in Figure 38. The takeoffs occur on runways 13 and 31 and landings on runway 22. This lowest altitude band shows the greatest changes in altitude over time and is explained by it being constituted of aircraft taking off and landing.

Mid-level patterns: The mid-level altitude band contains rotorcraft (red scatter points) and small fixed-wing aircraft (green scatter points). The rotorcraft activity follows a known rotorcraft air route. The figure also shows altitude change patterns, where a slight increase in altitude is observed for aircraft flying over the airport, possibly to maintain a bigger vertical separation from the LGA approach paths. These aircraft then descend back to approximately their initial altitude once clear of the airport.

High-altitude patterns: The highest altitude band shows primarily constant altitude operations, as is expected for high altitude commercial airliner en-route operations following mostly linear paths along air routes (yellow arrow). This altitude band is made up of mostly medium and large fixed wing aircraft (orange and blue scatter points) and also includes low volumes of small fixed wing aircraft (green scatter points).

The 3D airspace model of Figure 38 illustrates that, even for a small 5 × 5 × 5 km area and a 5-hour time interval, the airspace around LGA is complex and dynamic; characterized by multiple types of air vehicles, high density air traffic, and multiple types of operations (takeoffs, landings, airport flyover, high altitude air routes).

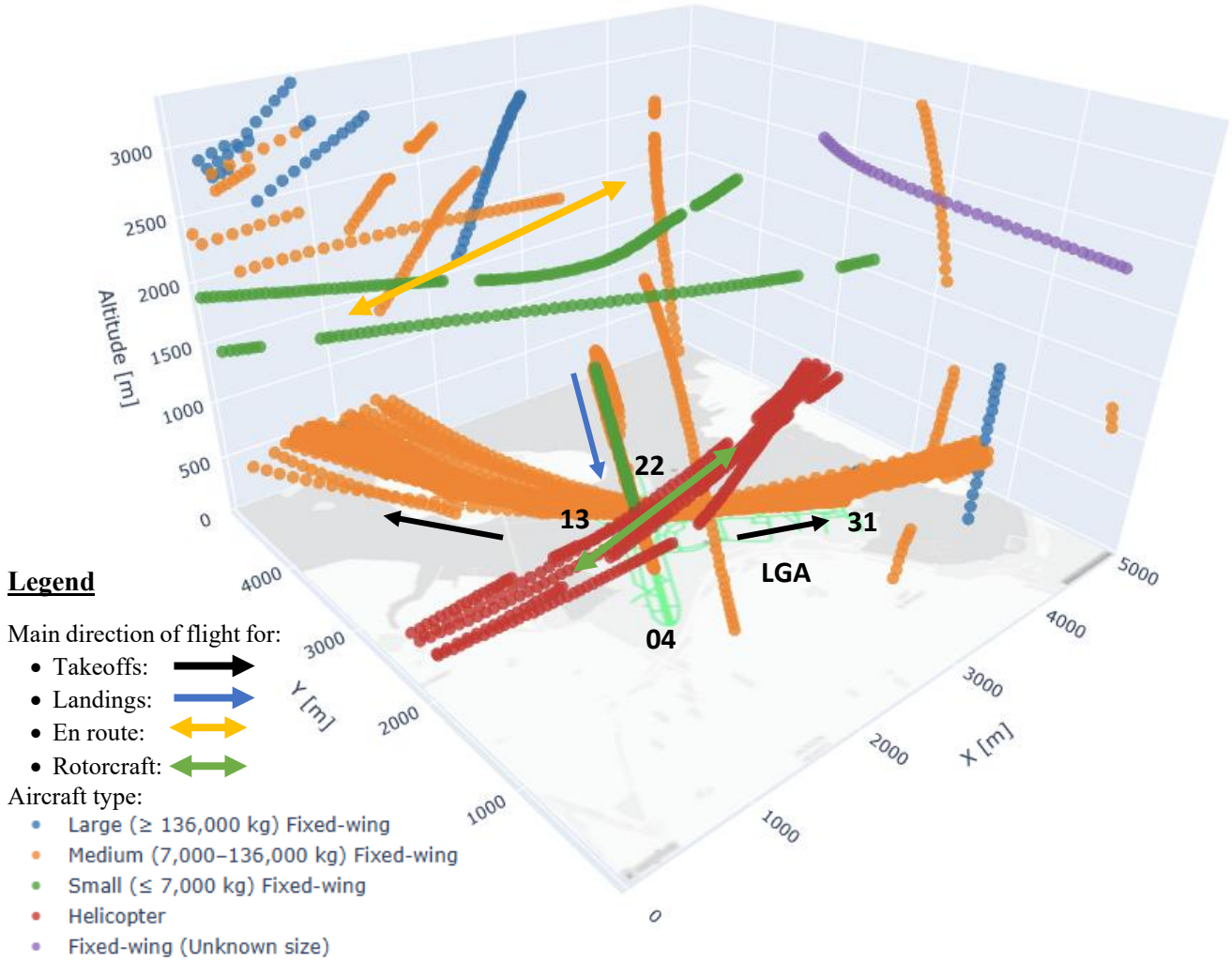


Figure 38: 3D map of air traffic data for Airspace model #1

5.3.1.3 Airspace model #1 weather data - 2D scatter plot

Figure 39 shows the 2D airspace model of cumulative weather radar data for a 1-hour subset of the 5-hour time range within the 5 × 5 × 5 km region centred on LGA airport. The figure depicts the storm cell centroids derived from raw NEXRAD level 2 reflectivity values, where each centroid corresponds to a localized convective structure identified and tracked within the DGGS grid. The storm cells directions of travel are identified by the arrows on each trajectory.

The figure shows that for a short time interval of 1 hour and a small 5 by 5 by 5 km region, 8 storm cells are captured by the airspace model. When compared to air traffic data, however, the movements of the 8 captured storm cells are much slower. Another observable difference is the more randomized (less structured) trajectories followed by the storm cells when compared to aircraft movement, which typically follow more predictable and linear patterns along known air routes or approach paths.

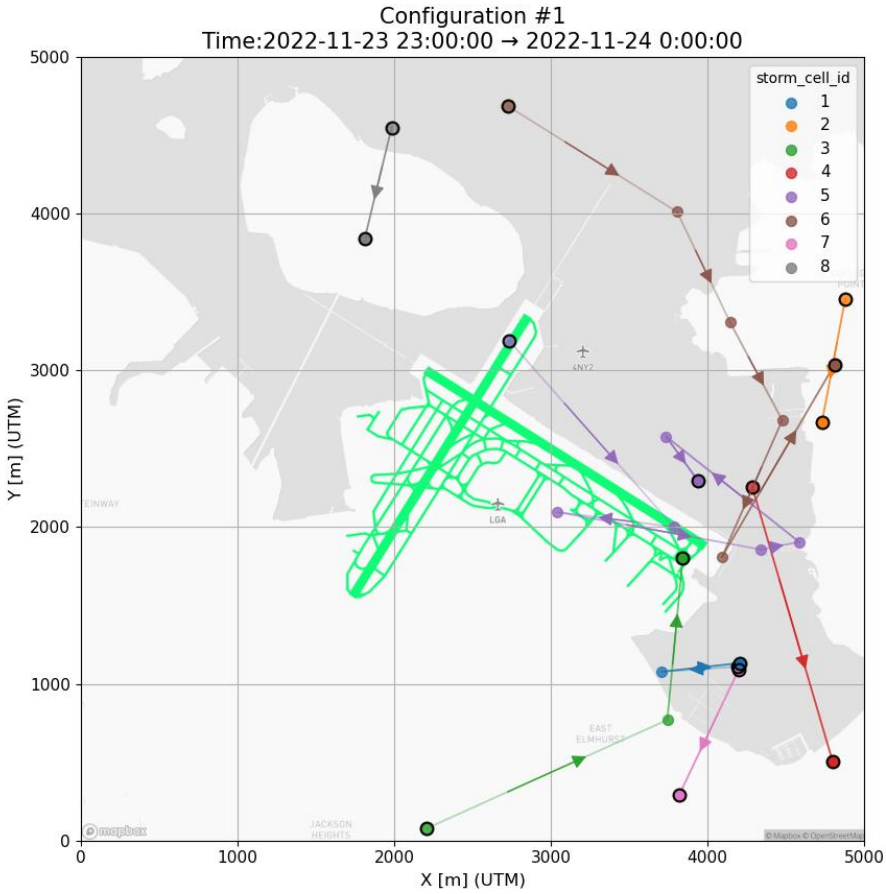


Figure 39: 2D map of weather data for Airspace model #1

5.3.1.4 Airspace model #1 weather data - 3D scatter plot

Figure 40 is the 3D representation of the data presented in Figure 39. The figure is used to highlight storm cell patterns observable across altitude layers.

The figure shows that most storm cells are in the 1000 to 2000 m altitude band. Some movement across altitude is also observed, although much less significant and slower when compared to air traffic altitude patterns.

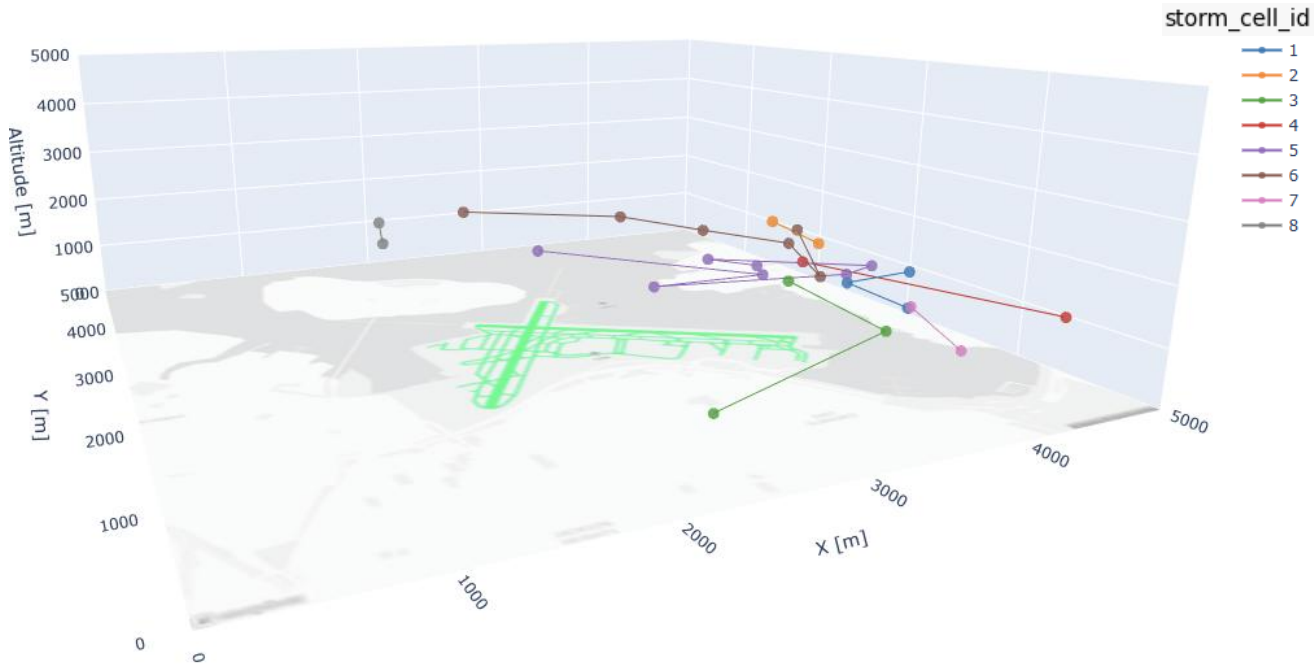


Figure 40: 3D map of weather data for Airspace model #1

5.3.2 Airspace model #2 results

5.3.2.1 Airspace model #2 air traffic - 2D scatter plot

Airspace Model #2 covers a substantially larger spatial region than Airspace Model #1. To maintain figure legibility, shorter representative subsets of the available regional data are used for the model-development visualizations in this section. Figures 41-42 show air traffic data for the 15-minute interval from 23:00 to 23:15 UTC (18:00 to 18:15 EST) on 22 November 2022, whereas Figures 43-44 show weather data for the 40-minute interval from 23:00 to 23:40 UTC (18:00 to 18:40 EST) on 23 November 2022. These shorter visualization intervals do not alter the spatial or DGGS configuration of Airspace Model #2. The computational evaluation in Section 5.4 evaluates the same model configuration over substantially larger historical-data workloads.

The 2D scatter plot of Figure 41 shows cumulative ADS-B data over the 15-minute time range for the 50 km by 50 km area of Airspace model #2. Each scatter point is an individual aircraft at a moment in time, totaling 274 unique aircraft over the 15 minutes (because of the high volume of traffic, many points overlap each other in the figure), and the colour of each scatter point corresponds to the geometric altitude of the aircraft. The four airports found in the region of interest are identified (JFK, LGA, EWR, and TEB) in the figure.

High density corridor structure near airports: The low-level air traffic trajectories (red scatter points) show multiple dense corridors converging toward and diverging from the three major New York airports—JFK, LGA, and EWR—indicating heavy terminal area operations typical of Class B airspace. The black arrows identify major outbound traffic (takeoffs) and the blue arrows identify major inbound traffic (landings) with respect to each airport, and there are clear inbound and outbound flow patterns aligned with the principal runway orientations of all four airports. The interweaving of these flows over the urban core (Midtown Manhattan region) illustrates the complexity of air-traffic management in this region.

Distinct altitude layer patterns: Three distinct groups of air traffic operations are observed based on altitude: a low-altitude layer (<2,000 m), an intermediate-altitude layer (~2,000–6,000 m), and a high-altitude layer (>6,000 m). The low-altitude layer (red–dark red trajectories) is dominated by aircraft operating within the terminal arrival and departure environment surrounding the major airports. The intermediate-altitude layer (yellow–orange trajectories) contains aircraft transitioning between terminal operations and higher-altitude flight, including climb and descent segments associated with arrivals and departures. The high-altitude layer (light blue–dark blue trajectories) is dominated by en-route overflights and higher-altitude arrival and departure segments crossing the region.

Distinct regional traffic interactions: Distinct regional traffic interactions can be observed at or between airports. TEB (Teterboro Airport), the only regional airport of the four airports, exhibits dense short, low-altitude routes, characteristic of business-jet operations and local general aviation patterns. LGA and JFK show structured arrival and departure paths extending east–west and southeast–northwest, respectively. EWR traffic interacts with TEB's and LGA's corridors, forming potential congestion nodes west of Manhattan where these routes cross at different altitudes.

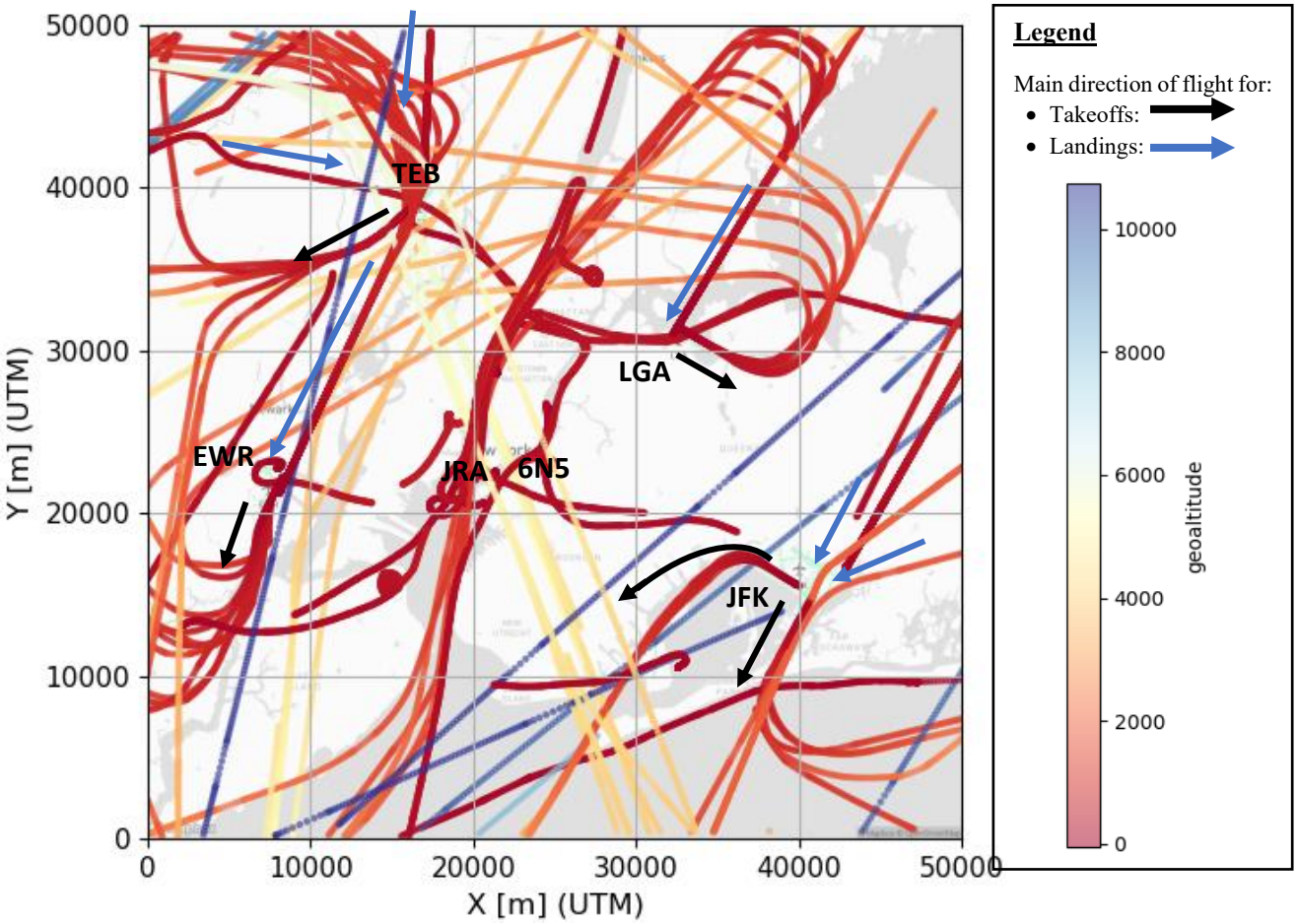


Figure 41: 2D map of air traffic data for Airspace model #2

5.3.2.2 Airspace model #2 air traffic - 3D scatter plot

The 3D airspace model of Figure 42 expands upon the 2D analysis of Figure 41 by introducing the vertical dimension to reveal the full spatial organization of aircraft trajectories within the New York airspace. Each scatter point is coloured based on aircraft type (either small, medium, large fixed-wing aircraft or rotorcraft) and the main directions of flight for takeoffs and landings near each airport are identified by the black and blue arrows. Several observations can be drawn from the figure:

Vertical separation of traffic layers: While the 2D view reveals three visually identifiable altitude layers, the 3D map makes the vertical separation explicit. The lowest layer (below ~2 000 m) forms dense, horizontally confined clusters surrounding JFK, LGA, EWR, and TEB. These correspond to Class B terminal sectors, with aircraft on final approach or initial climb segments. The intermediate layer (~2 000–6 000 m) features inclined trajectories, confirming climb and descent transition corridors connecting terminal and en-route altitudes and illustrating the spatial hand-off between tower/approach control and New York ARTCC. The upper layer (> 6 000 m, up to ~10 000 m) shows broad, near-level cruise paths transiting the region. In the 2D view of Figure 41, these appear as simple linear traces, but the 3D view confirms that they maintain altitude separation above terminal operations (above the Class B ceiling) and likely belong to Class E or upper-ARTCC controlled airspace below the 18 000 ft Class A boundary. This vertical depiction confirms the structured altitude hierarchy and clarifies how successive traffic layers interweave without direct crossing conflicts.

Inter-airport flow coordination across altitude: The 3D figure reveals how departure and arrival streams from multiple airports are vertically offset to prevent lateral conflict:

- Departures from JFK and LGA climb at distinct angles, merging only at higher altitudes (~4 000–6 000 m) where ARTCC sequencing occurs.
- EWR and TEB share overlapping corridors in the 2D view of Figure 41, but in 3D these flows are vertically staggered, demonstrating local deconfliction procedures over northern New Jersey.
- Cross-Manhattan flows appear separated by several hundred to a few thousand meters in altitude, aligning with published STAR (Standard Terminal Arrival Route) and SID (Standard Instrument Departure) profiles that manage multi-airport interactions.

Distinct altitude envelopes for helicopter versus fixed-wing operations: Helicopter operations (red scatter points) are a distinct class of low-altitude operations clearly visible in the dataset, which are primarily concentrated along the Hudson River (north–south corridor along Manhattan's west side), the East River (parallel corridor along the east side of Manhattan), and circling around key visual landmarks such as the Statue of Liberty. These operations are confined below ~600 m. The 3D rendering highlights how completely segregated these operations are from fixed-wing traffic, both laterally and vertically. This confirms the New York Special Flight Rules Areas (SFRA) design, where rotorcraft operate on dedicated low-altitude routes far below even the lowest fixed-wing arrivals. Multiple helicopter flights depart and land at the East 34th Street Heliport (6N5) and the West 30th Street Heliport (JRA), which are identified in the figure.

Improved comprehension of route geometry: Finally, the 3D model illustrates the trajectory curvature and climb gradients of departing and arriving aircraft that could not be inferred from 2D projections. It shows the steep climb angles for large and medium fixed-wing aircraft immediately after take-off from JFK and LGA and the shallower climb or descent profiles of business-jet operations around TEB and EWR.

In summary, the three-dimensional view of Airspace model #2 allows a clearer differentiation between overlapping layers of traffic and altitude-dependent route structures at a larger scale than in Airspace model

#1, which focused only of terminal operations. The figure highlights the complexity of the NYC airspace by depicting in the 3D model high density air operations, multi-international (JFK, EWR, LGA) and regional airport (TEB) air operation interactions, and varied air vehicle type operations.

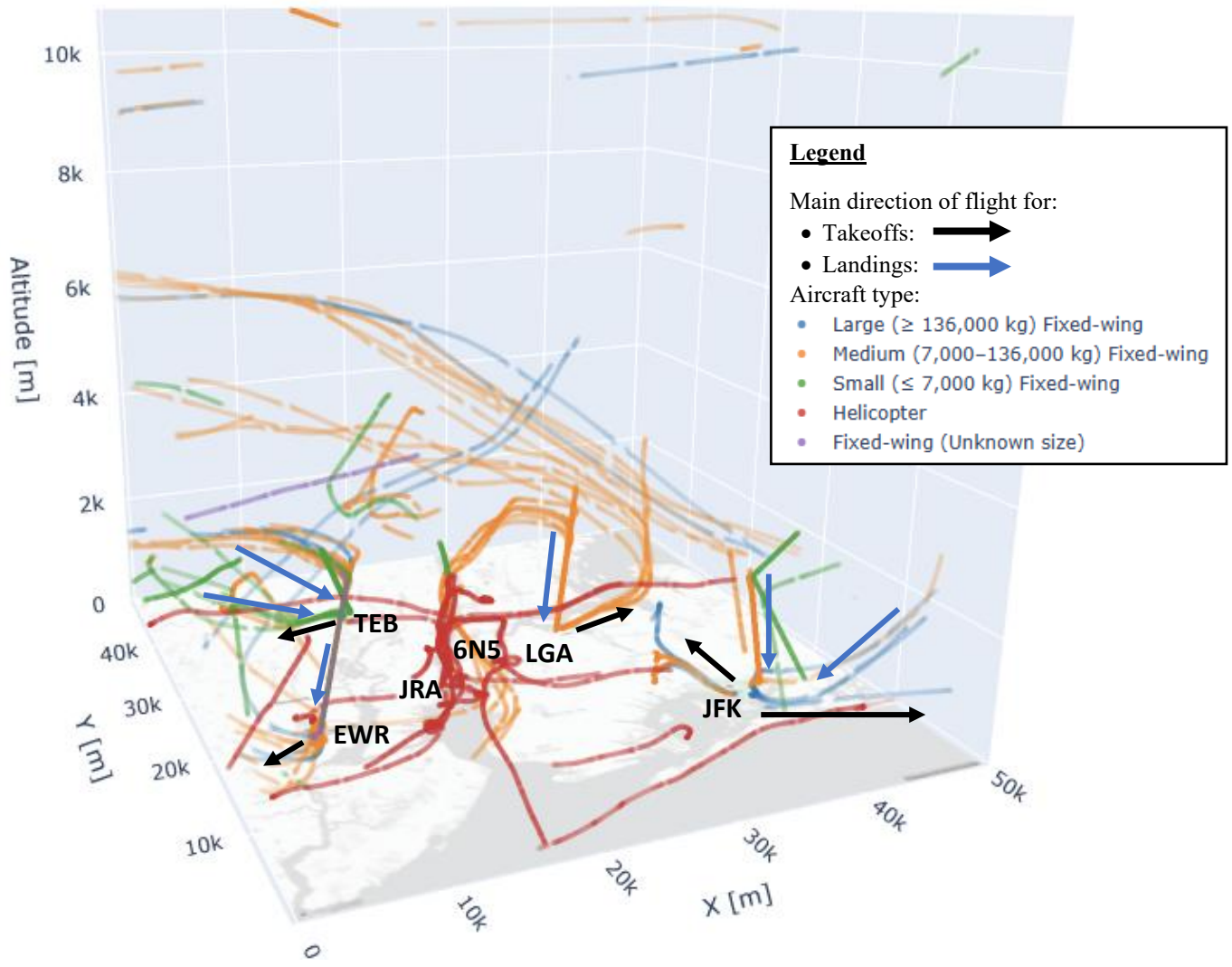


Figure 42: 3D map of air traffic data for Airspace model #2

5.3.2.3 Airspace model #2 weather data - 2D scatter plot

Figure 43 presents the two-dimensional weather representation for Airspace Model #2 over the 50 × 50 km regional domain for the 40-minute interval from 23:00 to 23:40 UTC (18:00 to 18:40 EST) on 23 November 2022. The figure shows the trajectories of 18 storm cells derived from the processed NEXRAD Level II radar observations. Each storm cell identifier is represented by a distinct colour, with the centroid positions connected in temporal order. The arrows superimposed on the trajectories indicate the direction of displacement of each storm cell during the selected interval. The Mapbox background provides geographic context relative to the New York metropolitan region and its major airports.

Compared with the smaller Airspace Model #1 domain, the larger regional model captures weather structures distributed across a substantially broader geographical area. Multiple storm cell trajectories are

visible around and between the principal airport regions, while several storm cells show successive centroid positions moving in a consistent direction over the selected interval. The figure therefore demonstrates that the same weather-processing and representation methodology used for Airspace Model #1 can also represent multiple time-varying weather entities across the larger Airspace Model #2 domain.

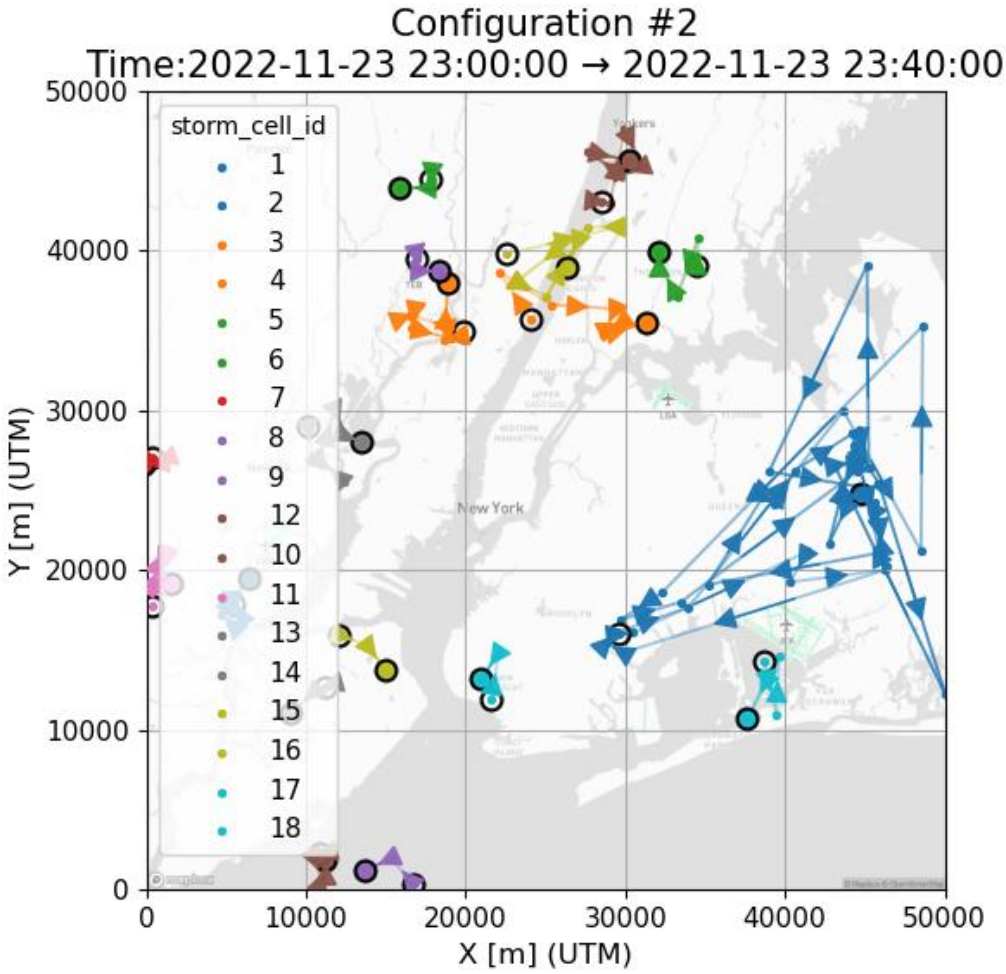


Figure 43: 2D map of weather data for Airspace model #2

5.3.2.4 Airspace model #2 weather data - 3D scatter plot

Figure 44 presents the same Airspace Model #2 storm cell data as Figure 43, but in three dimensions. The horizontal axes represent the 50 × 50 km regional model domain, the Mapbox surface provides the corresponding geographic reference, and the vertical axis shows the altitude of the derived storm cell centroids. Each storm cell retains the same colour identification used in Figure 44, allowing its horizontal and vertical evolution to be followed between successive observations.

The three-dimensional view shows that the identified weather structures occupy different altitude levels and that several storm cell trajectories exhibit changes in centroid altitude during the selected interval. This vertical variation is more readily distinguishable than in the two-dimensional representation and illustrates the additional information retained by the three-dimensional weather model. Together, Figures 44 and 45 verify that weather entities can be represented with time-varying horizontal position and altitude across the larger regional Airspace Model #2 domain.

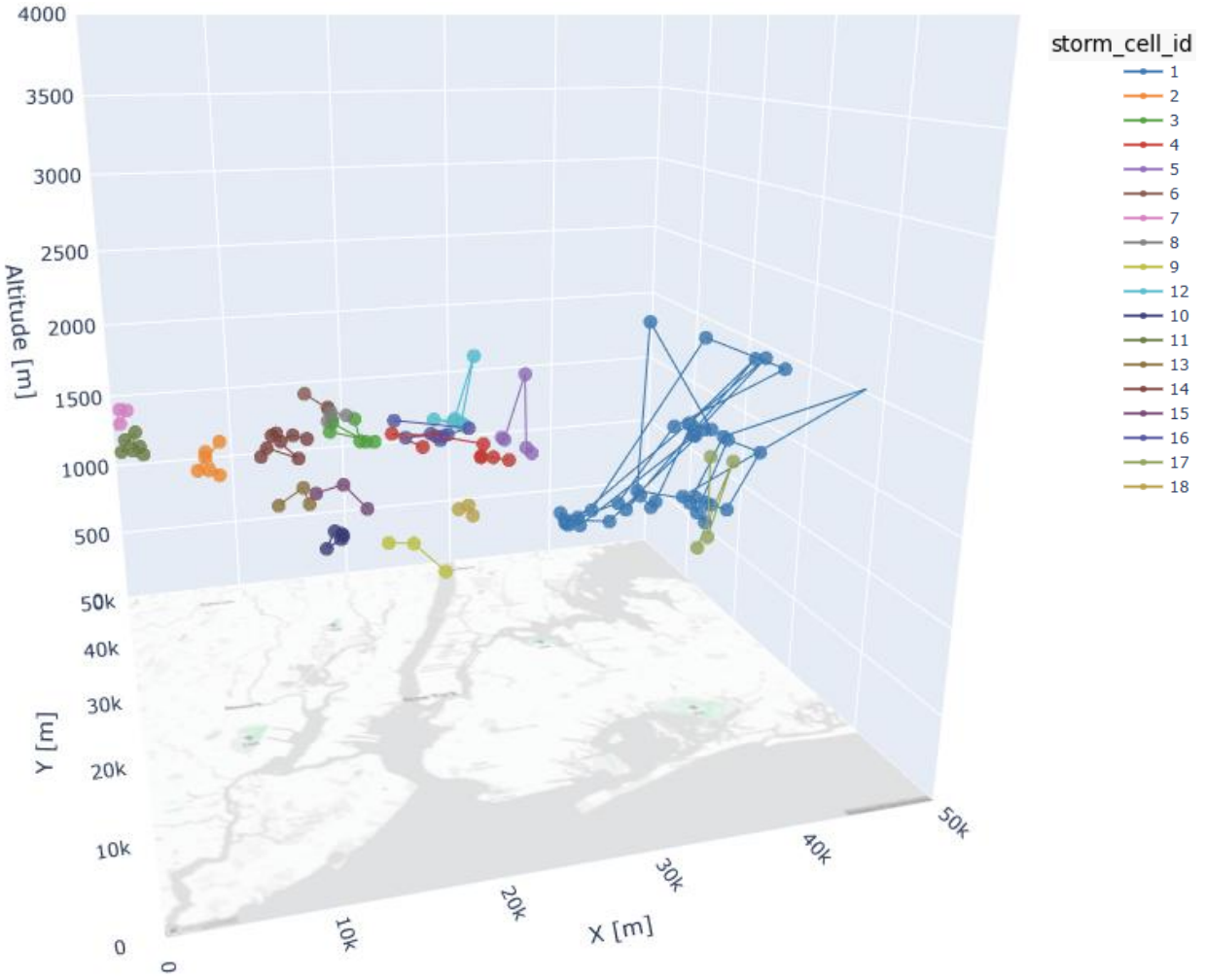


Figure 44: 3D map of weather data for Airspace model #2

5.3.2.5 Airspace model #2 weather data - 2D heat map

Figure 45 uses a 2D heat map visual to show information on the severity of each storm cell presented in Figure 43 and Figure 44 at four different altitude layers for each DGGS cell of Airspace model #2.

Identification of high severity storm cell hotspots: Figure 45 can be used to identify storm cell severity hotspots. The heat maps highlight that the most intense precipitation (30–50 dBZ; heavy to severe precipitation NOAA classes) occurred predominantly within the 1250–2500 m altitude range (Figure 45b), corresponding to the main convective layer observed over the region. The figure shows two hotspots where storm cells are more severe: one between LGA and JFK (the only yellow cell) and one near the south-western corner of the map in the Lower Bay (the only orange cell). These observations are consistent with known areas of storm cell activity near the coastal and inland transition zones, where convective cells merge and mature before dissipating further westward [465], [466]. Higher-altitude layers (> 2500 m) contain few or no cells during the analyzed interval.

Example use of a mixed-resolution DGGS model: Figure 45 also demonstrates cell-level aggregation within the mixed-resolution structure of Airspace Model #2. The regional model uses 5000 m base cells and 2500 m refined cells around selected airport regions. The finer cells provide greater spatial detail in

these areas while the coarser base grid limits the number of cells required across the remainder of the 50 km × 50 km domain. The figure therefore illustrates how the same weather variable can be represented at different spatial resolutions within one DGGS-based airspace model.

These results complement those of Airspace model #1 by showing how the DGGS-based airspace model effectively aggregates and visualizes meteorological data over larger spatial extents, capturing regional scale convective organization while retaining cell-level detail

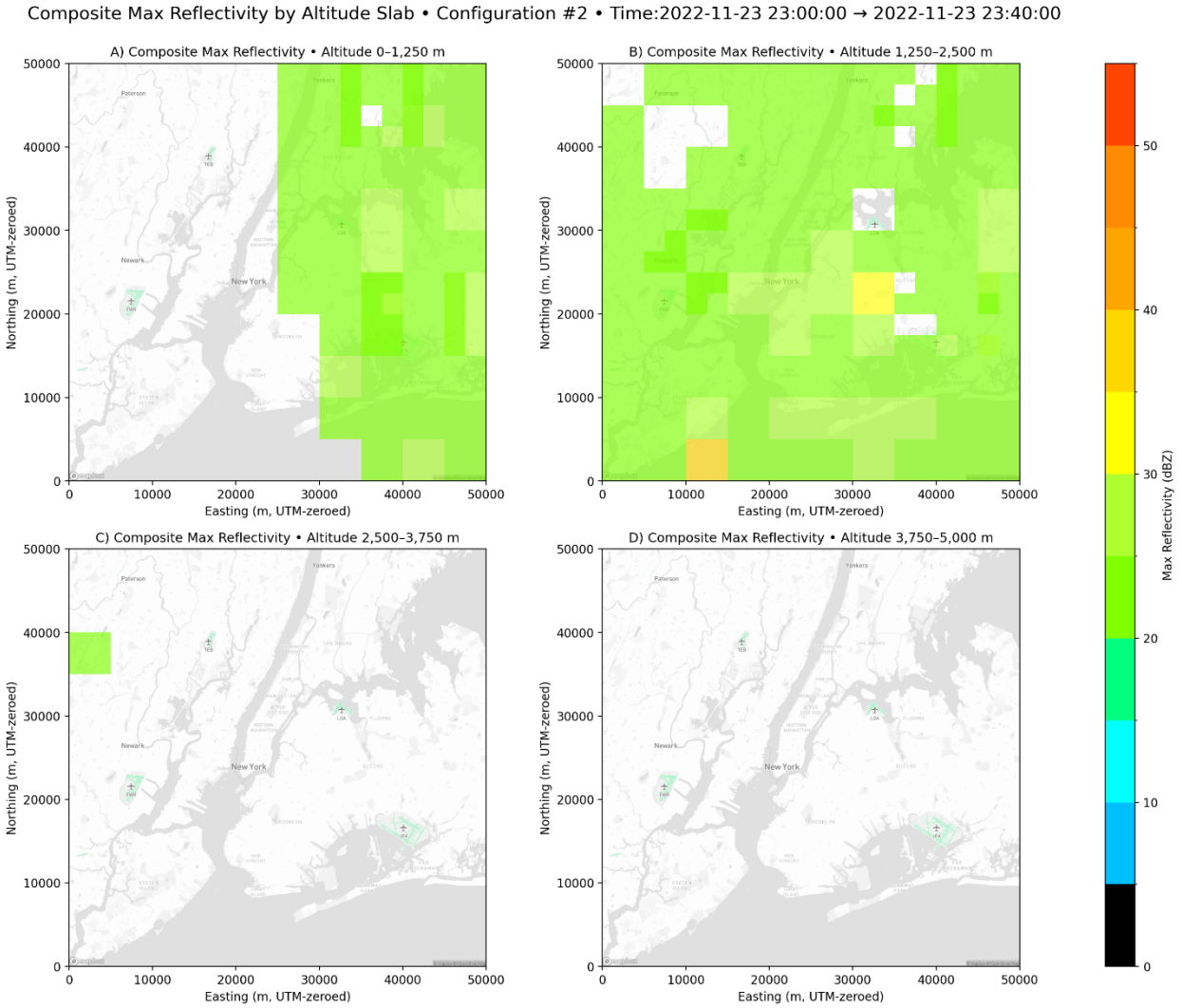


Figure 45: Composite maximum reflectivity aggregation for Airspace model #2

The results in this section demonstrate that the developed workflow produces two DGGS-based 4D airspace models that integrate aircraft trajectory data and weather radar data within a common spatial and temporal framework. The figures are used as model-verification outputs rather than as independent analytical results. They verify that the source datasets were correctly processed, georeferenced, mapped into the DGGS structure, and represented in both two-dimensional and three-dimensional visual forms.

Table 20 summarizes the evidence presented in the results of Section 5.3 for each model-development validation requirement defined in Section 5.1.

Table 20: Summary of the results and their corresponding validation requirement

Validation requirement	Evidence shown in results
Integrates aircraft data	ADS-B trajectories are processed, georeferenced, mapped to the model domain, and visualized in 2D and 3D.
Integrates weather data	Radar-derived storm cell data are processed, georeferenced, mapped to the same DGGS framework, and visualized in 2D and 3D.
Supports 3D representation	Aircraft and weather outputs are represented across horizontal position and altitude.
Supports time-varying data	Aircraft and weather entities retain time information and are shown over selected analysis intervals.
Supports different model scales	Airspace model #1 and Airspace model #2 are generated using the same workflow but different spatial extents and DGGS resolution strategies.
Supports cell-level aggregation	Weather intensity is aggregated at the DGGS cell level across altitude layers.

The results in this section establish the model-development component of Experiment #1. They demonstrate that aircraft and weather data can be represented within the common DGGS-based spatial and temporal structure used by the thesis. They do not, by themselves, establish the computational efficiency of generating that representation; the computational evaluation is presented separately in Section 5.4.

5.4 Computational evaluation results

This section presents the computational evaluation results for Experiment #1 using the six historical workloads and four processing scenarios defined in Section 5.1. The spatial extent and DGGS configuration are held constant using Airspace Model #2 while the duration and composition of the historical source data increase from TR1 to TR6. Results are summarized using the median wall-clock time, mean resident set size (RSS), and peak RSS across four valid repetitions of each condition. Saved model-output file size and LOAD execution time are evaluated separately to quantify storage and model-reuse behaviour.

5.4.1 Workload characterization

The six historical workloads used in the computational evaluation have been presented in Table 14 of Section 5.1. The duration increases from 1 h for TR1 to 48 h for TR6, while the total size of the corresponding raw ADS-B and radar source files increases from 0.5 GiB to 17.7 GiB. The TR6 raw source-data size therefore slightly exceeds the 16 GB of physical memory installed in the test computer. This does not mean that 17.7 GiB must be simultaneously loaded into memory: the value represents the combined size of the source files selected for the TR6 BUILD interval, whereas the processing workflow reads and transforms subsets of those source data during different stages of execution.

Workload composition also changes substantially across TR1–TR6, where "*workload composition*" refers to the number of aircraft observations, trajectory segments, radar observations, and derived weather records contained within each historical interval. For example, TR1 contains 0.7 million ADS-B source observations, 3,857 trajectory segments, and 842 final radar points, whereas TR6 contains 23.5 million ADS-B observations, 113,268 trajectory segments, and 454,502 final radar points. These quantities do not increase proportionally with elapsed time because aircraft traffic and weather activity vary over the selected

48-hour period. Computational behaviour must therefore be interpreted using both nominal workload size and the actual aircraft and weather information contained within each interval.

5.4.2 BUILD mode runtime results

Figure 46 shows the median BUILD wall-clock time for the four processing scenarios across TR1–TR6. For the S1 baseline, median BUILD time increases from 3.9 min for TR1, which processes 0.5 GiB of raw source data, 3857 ADS-B trajectories, and 842 retained radar points, to 129.9 min for TR6, which processes 17.7 GiB of source data, 113268 ADS-B trajectories, and 454,502 retained radar points. Runtime does not increase proportionally with historical duration because the amount and composition of aircraft and weather information also change between workloads.

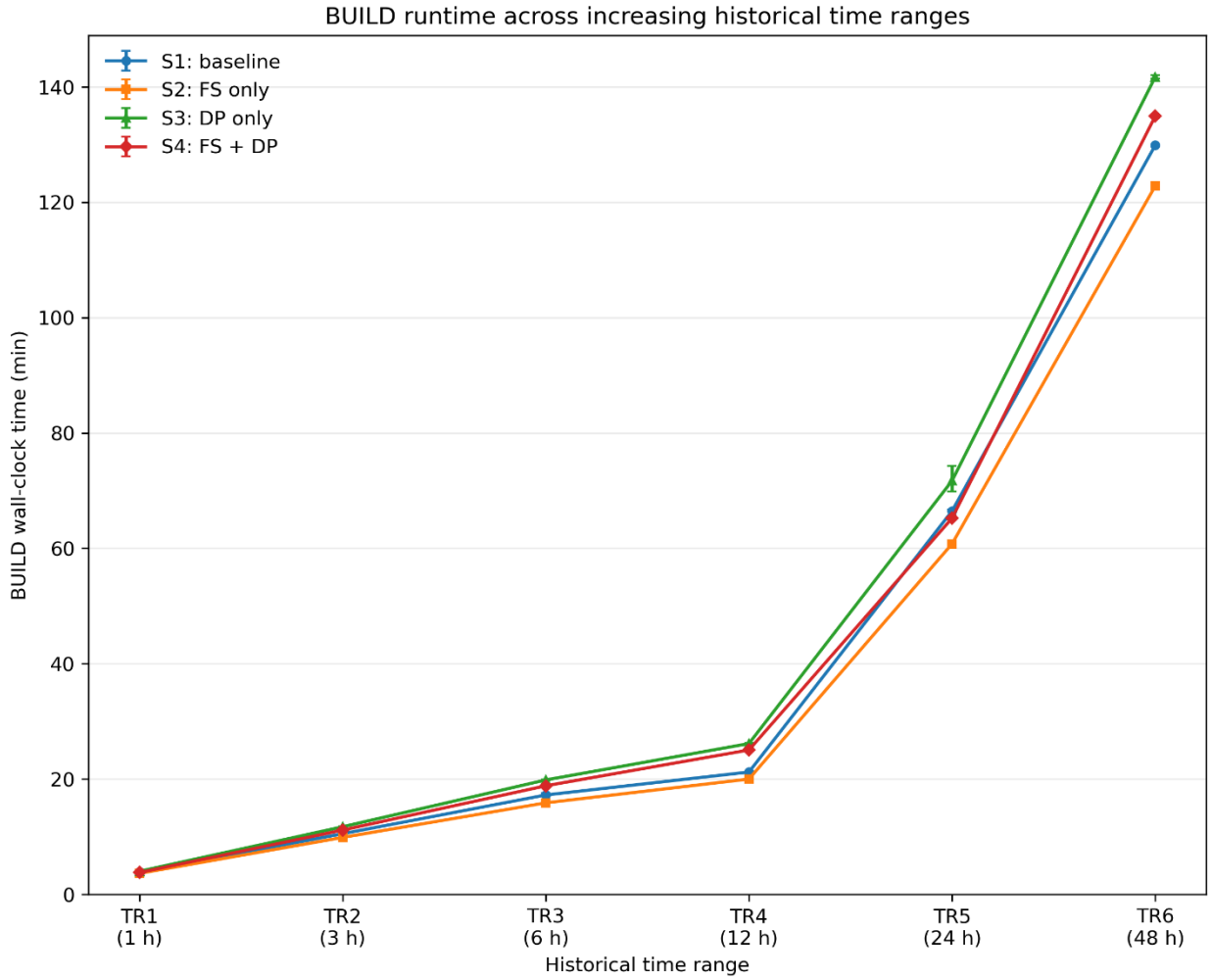


Figure 46: BUILD runtime across increasing workloads

Feature selection produces the most consistent BUILD-time reduction. Relative to S1, S2 reduces median BUILD time by 6.4%, 6.1%, 7.9%, 5.7%, 8.6%, and 5.4% for TR1 through TR6, respectively. This improvement is smaller than the reduction in the number of retained variables because many time-consuming operations still depend on the number of aircraft trajectories and radar observations being processed. Aircraft observations must still be filtered, segmented, transformed, and mapped to the DGGS structure, while radar observations must still be filtered, transformed, and processed into the required

weather outputs. FS therefore reduces the cost of carrying variables that are not required by the intended downstream analyses, but does not remove these observation-level processing operations.

Dataset partitioning has the opposite runtime effect. Relative to S1, the S3 scenario changes median BUILD time by +3.0%, +11.5%, +15.1%, +23.3%, +7.9%, and +9.1% for TR1 through TR6, respectively. When DP is enabled, source data are accessed and stored in multiple temporal partitions, temporary partition outputs are written and read, and the reduced data are subsequently recombined before global processing steps are completed. These additional input/output and recombination operations increase total execution time. DP is therefore not a runtime optimization in this experiment; its principal benefit is reduced memory demand, as discussed in Section 5.4.3 below.

The combined S4 strategy reflects both effects. Its BUILD-time change relative to S1 ranges from a small reduction for TR1 and TR5 to an increase for the remaining workloads. At TR6, S4 increases median BUILD time by approximately 4%. These results show that no single FS/DP scenario minimizes execution time for every workload.

5.4.3 Memory demand and the role of DP

Process memory is evaluated using two complementary metrics. **Mean RSS** represents the average amount of physical memory occupied by the process during execution and is used here as an indicator of sustained memory demand. **Peak RSS** is the maximum observed resident memory during an execution and represents the highest instantaneous memory demand placed on the computer. Mean RSS therefore indicates how much memory the workflow typically occupies, whereas peak RSS indicates the memory headroom required to accommodate its most demanding processing stage. A workflow can consequently exhibit a relatively low mean RSS while still producing a short-duration peak that determines whether the computation is feasible on memory-constrained hardware. Figures 47 and 48 present these two metrics separately.

Figure 47 shows median mean RSS for BUILD execution. Dataset partitioning reduces sustained memory demand across all six workloads. Relative to S1, S3 reduces mean RSS by 19.2% for TR1, 27.5% for TR2, 45.7% for TR3, 17.6% for TR4, 58.8% for TR5, and 42.2% for TR6. The combined S4 strategy follows the same overall pattern, showing that the memory benefit of dataset partitioning is preserved when feature selection is enabled. At TR6, S4 reduces mean RSS by 52.6%, from 5.8 GiB for S1 to 2.8 GiB. The S3 and S4 results therefore show that dataset partitioning, rather than feature selection alone, is the principal mechanism reducing sustained memory demand during BUILD.

Figure 48 shows the corresponding peak RSS values. The effect of DP on maximum instantaneous memory demand varies more strongly with workload composition. S3 reduces peak RSS by 51.2% for TR3, 55.9% for TR5, and 47.0% for TR6, while the corresponding S4 reductions are 46.6%, 55.8%, and 54.6%. At TR6, median peak RSS decreases from 12.308 GiB for S1 to 5.590 GiB for S4. The benefit is much smaller for TR1 and TR4, indicating that the short-duration operation responsible for peak memory in those workloads is not strongly affected by partitioning. DP therefore provides its largest practical advantage when the active aircraft or radar data processed by memory-intensive stages becomes large.

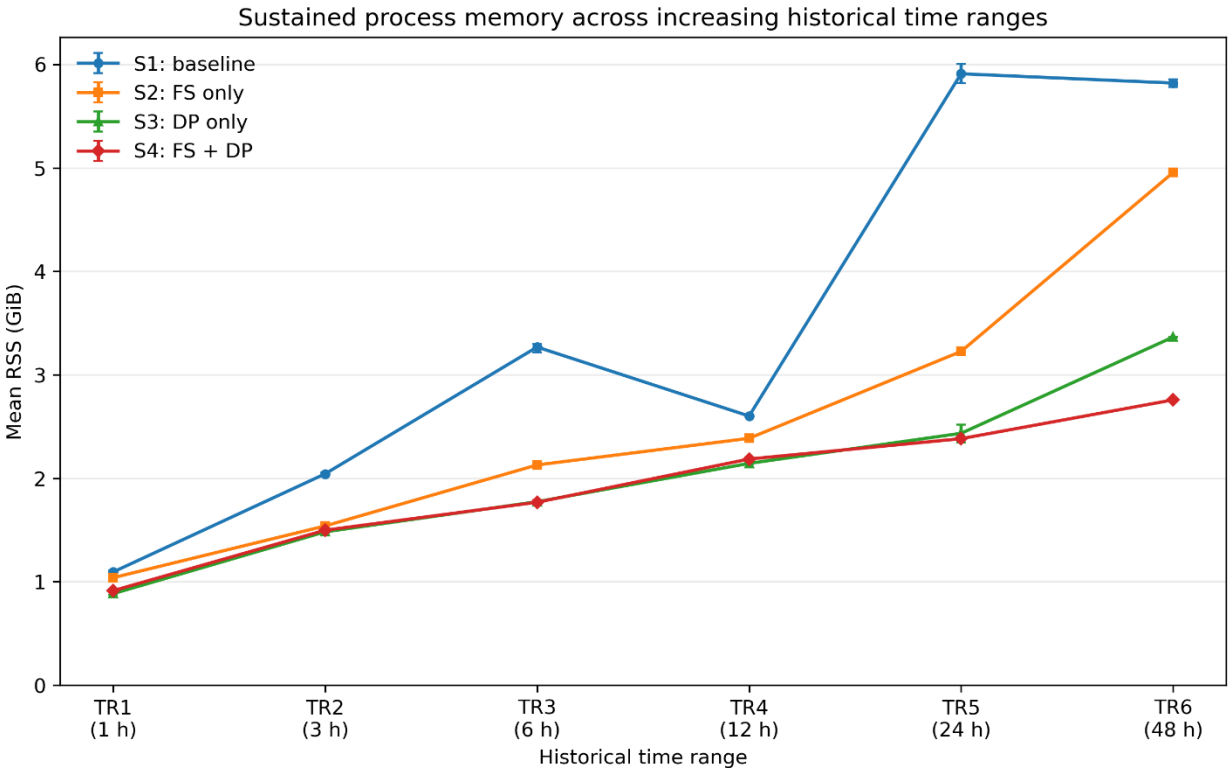


Figure 47: Mean RSS scaling

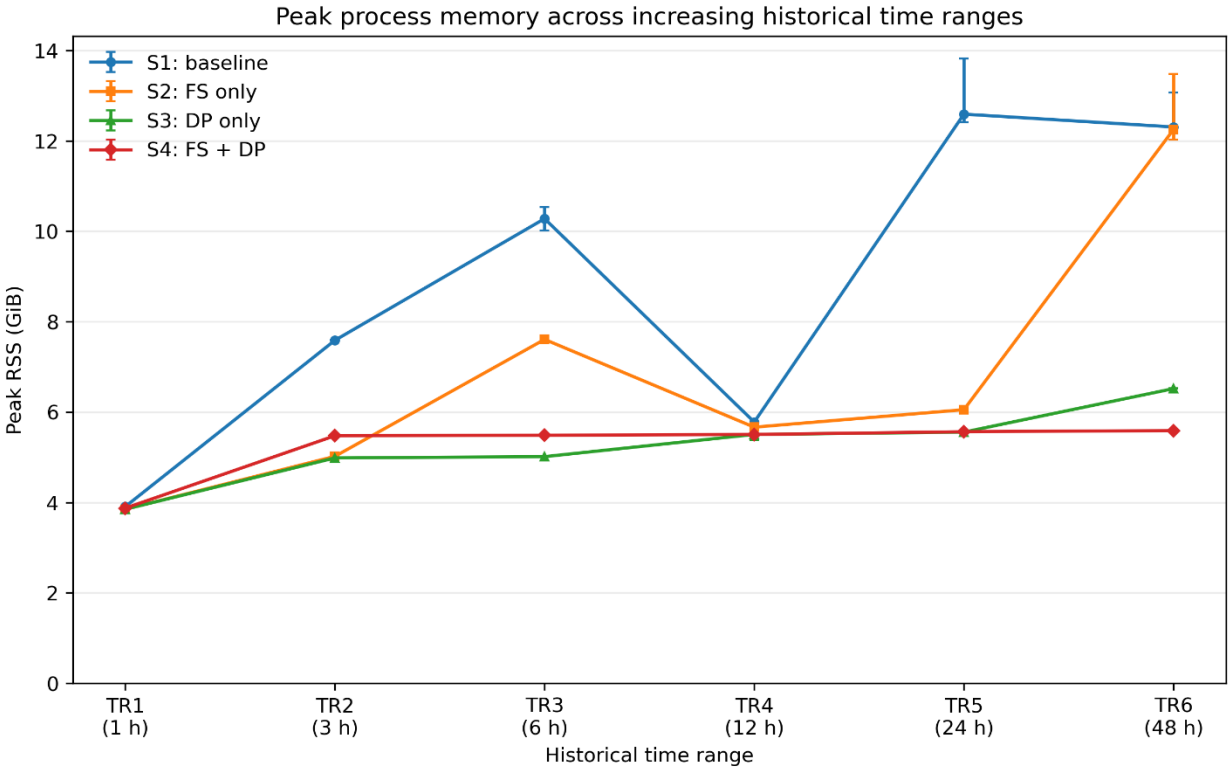


Figure 48: Peak RSS scaling

Feature selection alone also reduces memory use, although its effect on peak RSS varies with workload composition. This indicates that reducing dataframe width can lower sustained memory demand without necessarily removing the short-duration processing operations responsible for the maximum memory peak.

5.4.4 The effect of feature selection on model output size

Figure 49 reports the sizes of the processed ADS-B and weather model files produced during BUILD and subsequently read during LOAD. These values are distinct from the raw source-data sizes reported in Table 14. They represent the saved, analysis-ready model outputs available for reuse after model construction.

Feature selection substantially reduces these saved file sizes. The baseline ADS-B representation contains 17 source variables. The FS-enabled workflow uses seven during processing and writes five to the saved ADS-B model output, corresponding to a **70.6% reduction in final variable count**. The weather representation is reduced from ten source variables to five saved variables, corresponding to a **50.0% reduction** in final variable count.

As shown in Figure 49, the reduction in file size is consistent across workloads. S2 reduces the saved ADS-B output size by 63.9%, 63.9%, 63.9%, 63.9%, 63.6%, and 66.3% from TR1 through TR6, respectively. The corresponding reductions in the saved weather output are 60.9%, 61.7%, 62.0%, 61.9%, 61.6%, and 61.5%. At TR6, for example, the saved ADS-B output decreases from **745.3 MB to 250.9 MB**, while the saved weather output decreases from **68.9 MB to 26.5 MB**.

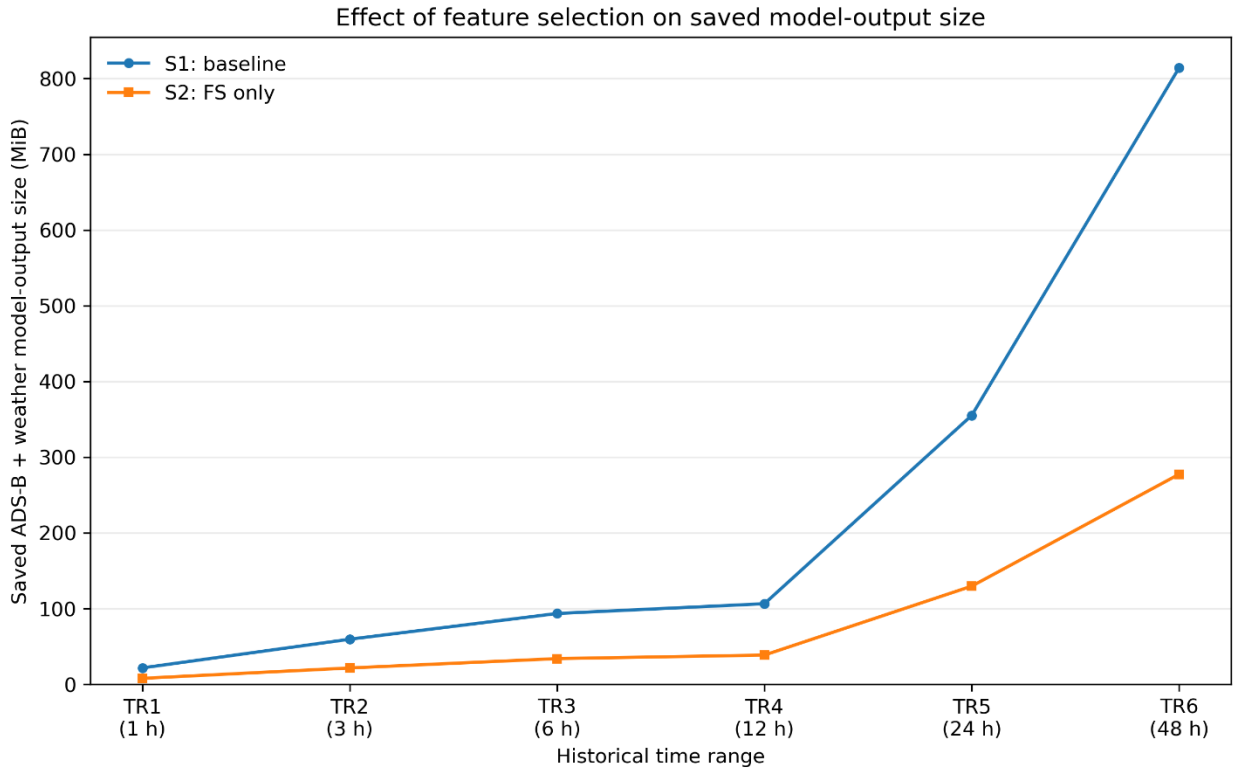


Figure 49: Effect of feature selection on saved model-output size

Feature selection therefore provides three related computational benefits: a modest reduction in BUILD runtime, a substantially larger reduction in saved model-output size, and a reduction in the amount of processed data that must subsequently be read during LOAD execution.

5.4.5 BUILD versus LOAD and reusable model value

Figure 50 compares the computational cost of constructing and subsequently reusing the airspace model across TR1–TR6. BUILD time increases from minutes to more than two hours as the historical time range grows, whereas LOAD remains below approximately 12 s even at TR6. At TR6, median S1 execution decreases from 129.9 min in BUILD to 11.382 s in LOAD, corresponding to a **99.9% reduction**. A logarithmic y-axis is used because the construction and reuse phases operate on substantially different time scales; on a linear scale, the LOAD curves would collapse near zero and their variation would be difficult to distinguish. The result shows that the computationally expensive source-data ingestion, preprocessing, georeferencing, and DGGS mapping are incurred during model construction, whereas subsequent reuse requires only access to the previously generated model outputs.

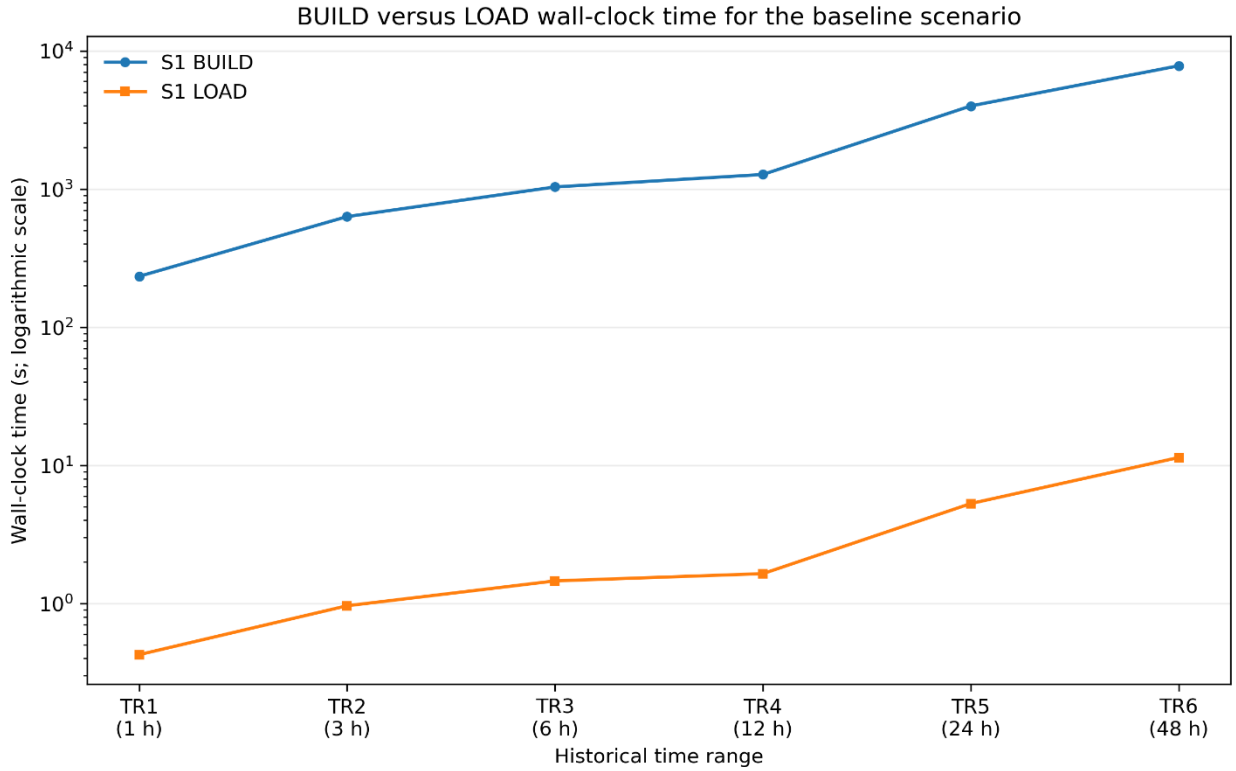


Figure 50: Cost of building versus reusing the baseline airspace model

Figure 51 isolates the LOAD execution mode and compares all four processing scenarios. S1 and S3 remain closely grouped, whereas S2 and S4 form a lower-runtime group as the historical time range increases. At TR6, median LOAD times are 11.4 s for S1, 3.6 s for S2, 11.1 s for S3, and 4.4 s for S4. The similarity between S1 and S3, and between S2 and S4, indicates that feature selection rather than dataset partitioning is the principal factor controlling model-reuse time. This is consistent with Figure 50: feature selection reduces the size of the saved ADS-B and weather model outputs that must subsequently be read from storage, whereas dataset partitioning primarily changes how source data are processed during BUILD and provides little additional benefit once the final outputs already exist.

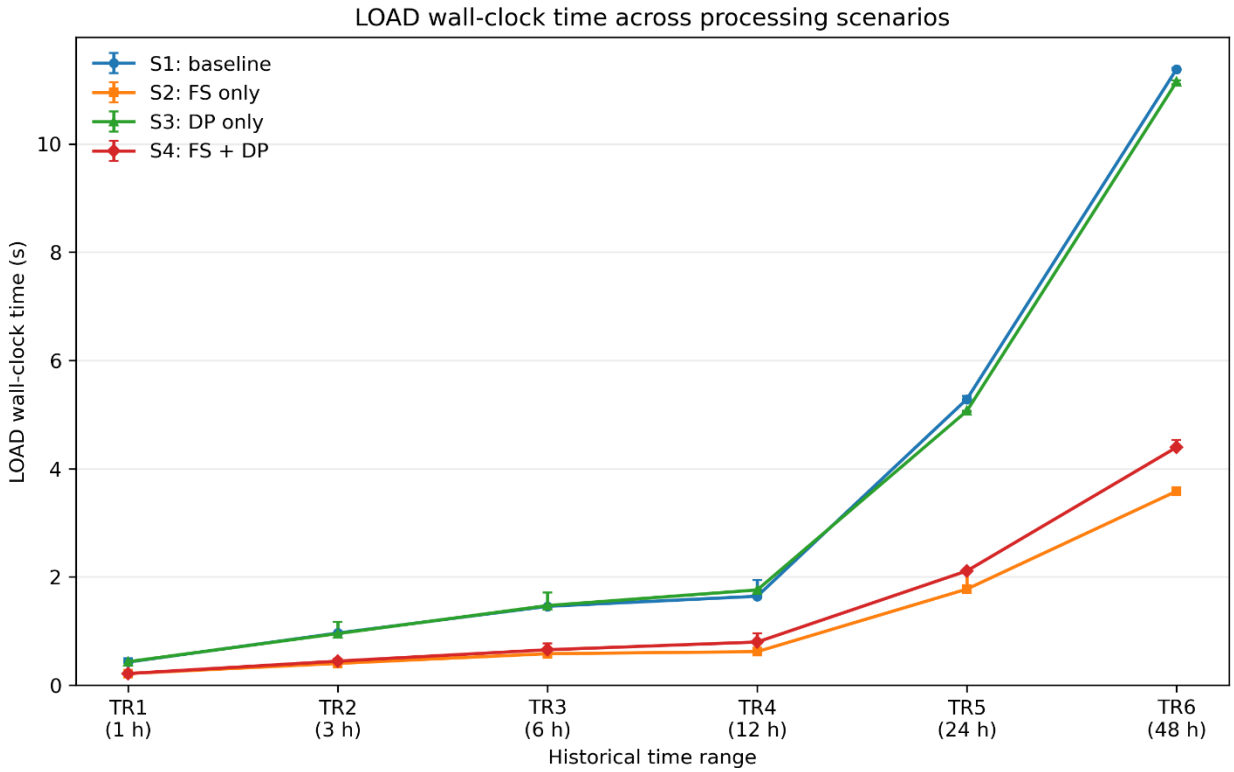


Figure 51: Model reuse time across scenarios

The BUILD–LOAD difference is evident under every processing scenario. At TR6, the BUILD-to-LOAD wall-clock reductions are 99.85% for S1 (11.382/7,792.44 s), 99.95% for S2 (3.583/7,368.84 s), 99.87% for S3 (11.148/8,502.72 s), and 99.95% for S4 (4.394/8,095.62 s), where the values in parentheses represent the LOAD-to-BUILD time ratios. BUILD and LOAD should therefore be interpreted as different phases of the airspace-model lifecycle rather than as competing optimization strategies.

This separation provides direct evidence of model reusability. After one BUILD execution transforms the heterogeneous source datasets into processed and DGGS-indexed model outputs, the resulting model can be reloaded for subsequent analytical operations without reconstructing the underlying historical airspace representation. Examples include evaluating alternative risk formulations or parameter settings on the same airspace interval, generating additional visualizations or derived metrics, and performing multiple clustering analyses using the same processed aircraft trajectories.

Taken together, the computational results identify distinct roles for FS and DP. Feature selection consistently decreases BUILD runtime and substantially decreases saved model-output size. Dataset partitioning generally increases BUILD runtime but can substantially reduce both sustained and peak memory demand when larger amounts of aircraft or weather data are active during memory-intensive processing stages. The combined S4 strategy is therefore most useful when memory availability is a stronger constraint than minimum execution time. Finally, LOAD separates the one-time cost of model construction from repeated analytical use, reducing subsequent access time from minutes or hours to seconds.

5.4.6 Airspace model output consistency

Output consistency was evaluated separately from computational performance to verify that changes in runtime or memory demand were not obtained by changing the analytical result produced by the model. First, every valid benchmark execution was required to produce the expected workload-specific numbers of final ADS-B observations, radar observations, and storm-centroid records. An execution that did not satisfy these completeness criteria was not accepted as one of the four valid repetitions. Second, where matched scenario outputs were retained, canonicalized output comparisons were used to verify the analytical fields required by subsequent analyses, including DGGS cell identifiers, Morton codes, trajectory information, spatial coordinates, and altitude values. The retained comparisons were identical except for one TR3/S4 ADS-B comparison in which 47 stored latitude or longitude values differed by a single 0.0001° rounding increment; the corresponding DGGS cell identifiers and Morton codes were unchanged. These checks provide evidence that FS and DP altered the computational execution of the workflow without materially altering the model outputs required by downstream analyses.

5.5 Experiment #1 discussion

Experiment #1 provides evidence for both parts of Research Question #1 and Hypothesis #1. The model-development results show that historical ADS-B aircraft trajectories and NOAA NEXRAD radar observations can be collected, processed, transformed into a common geospatial reference system, associated with three-dimensional DGGS cells and time, and saved as reusable model outputs. Airspace Models #1 and #2 demonstrate this process under different spatial extents and DGGS resolution strategies. Their 2D and 3D representations preserve recognizable airport-centred trajectory patterns, altitude-dependent traffic structure, weather distributions, and DGGS cell-level organization. These observations provide qualitative evidence that the model representation remains interpretable in the context of the underlying airspace data; they are not presented as an independent quantitative validation of operational accuracy.

Feature selection provides the most consistent reduction in BUILD runtime. Across TR1–TR6, S2 decreases median BUILD time by 5.4–8.6% relative to S1 while reducing the saved ADS-B and weather model-output sizes by approximately 61–66%. This result shows that eliminating variables not required by the intended downstream analyses reduces both model-generation cost and the amount of processed data that must be stored and reused. The smaller runtime reduction relative to the file-size reduction is expected because FS does not eliminate the trajectory and radar observations on which the principal processing operations are performed.

Dataset partitioning addresses a different computational constraint. S3 increases BUILD runtime for every workload because partitioned processing requires additional source reads, temporary writes and reads, and recombination of reduced partition outputs. These additional operations increase execution time but can substantially reduce memory demand. The results therefore identify DP primarily as a memory-management strategy rather than a runtime optimization.

The results also show that computational demand depends on the composition of the historical data, not only on the duration or region size of the selected source files. Aircraft activity and weather activity vary substantially across the six nested time intervals, producing non-proportional changes in aircraft observations, trajectory segments, radar observations, and derived storm information. The computational scaling of the airspace model is consequently workload-dependent: two historical intervals of different duration can impose similar or different computational demands depending on the observations contained within them.

BUILD and LOAD provide the final evidence of model reuse. BUILD represents the one-time transformation of source aircraft and weather data into the processed DGGS-based representation, whereas LOAD accesses the previously generated outputs. For TR6, S1 requires 129.874 min to build and 11.382 s to load; S4 requires 134.927 min to build and 4.394 s to load. The saved model can therefore be reused repeatedly for subsequent analytical operations without repeating the complete source-data processing workflow.

Taken together, the computational results support a clear operating strategy for the implemented workflow. **S2, with feature selection enabled and dataset partitioning disabled, is preferred when available memory is sufficient because feature selection provides the most consistent reduction in BUILD time and saved model-output size without the additional processing overhead introduced by dataset partitioning. S4, with both feature selection and dataset partitioning enabled, is preferred when memory demand becomes a limiting constraint because it retains the benefits of feature selection while substantially reducing sustained and peak memory usage, at the cost of additional BUILD-time overhead.** Once the required reusable model outputs have been generated, LOAD should be used for repeated downstream analyses whenever regeneration is unnecessary. S1 and S3 therefore remain useful baseline and comparison scenarios rather than the preferred operating configurations for the downstream analytical workflow evaluated in this thesis.

The outputs of Experiment #1 form the common data foundation for the remaining thesis experiments. Chapter 6 presents the use of the DGGS-based model to derive location-based airspace risk metrics, while Chapter 7 describes the use of the processed aircraft trajectories for behaviour-based categorization using unsupervised machine learning.

Chapter 6: DGGS-based airspace risk analysis

Research Question #2 asks how a reusable 4D airspace model can be used to derive and interpret multiple types of location-based airspace metrics from heterogeneous, time-varying data. This chapter presents Experiment #2, which evaluates the **location-based metric computation branch** of the proposed methodology through a location-based airspace risk case study. Three related analytical products are derived: hazardous-weather risk, location-based midair-collision (MAC) risk, and combined weather–MAC risk. Each product is associated with common three-dimensional DGGS locations and time steps. The experiment therefore tests Hypothesis #2 and provides a controlled case study of the broader analytical capability addressed by Research Question #2.

The risk-analysis methodology developed in Experiment #2 builds on earlier work by Vincent-Boulay and Marsden. An initial location-based MAC risk formulation using ADS-B data was presented in [5]. This work was subsequently expanded in [2] into a DGGS-based geospatial airspace risk methodology incorporating aircraft and weather data. The present chapter develops these earlier studies within the broader thesis framework by treating weather risk, MAC risk, and their combination as derived location-based analytical products generated from the reusable 4D airspace model developed in Experiment #1.

Figure 52 shows the relationship between the methodology steps and Experiment #2. The experiment's input is the reusable DGGS-based 4D airspace model developed in Experiment #1, including processed aircraft trajectory data, processed weather-radar and storm cell data, DGGS cell centroids, and time-indexed model outputs. The experiment applies additional analytical computations to these existing model contents rather than reconstructing a separate airspace representation for each risk component. The resulting outputs are location-based weather, MAC, and combined risk indicators that share the same spatial and temporal reference structure.

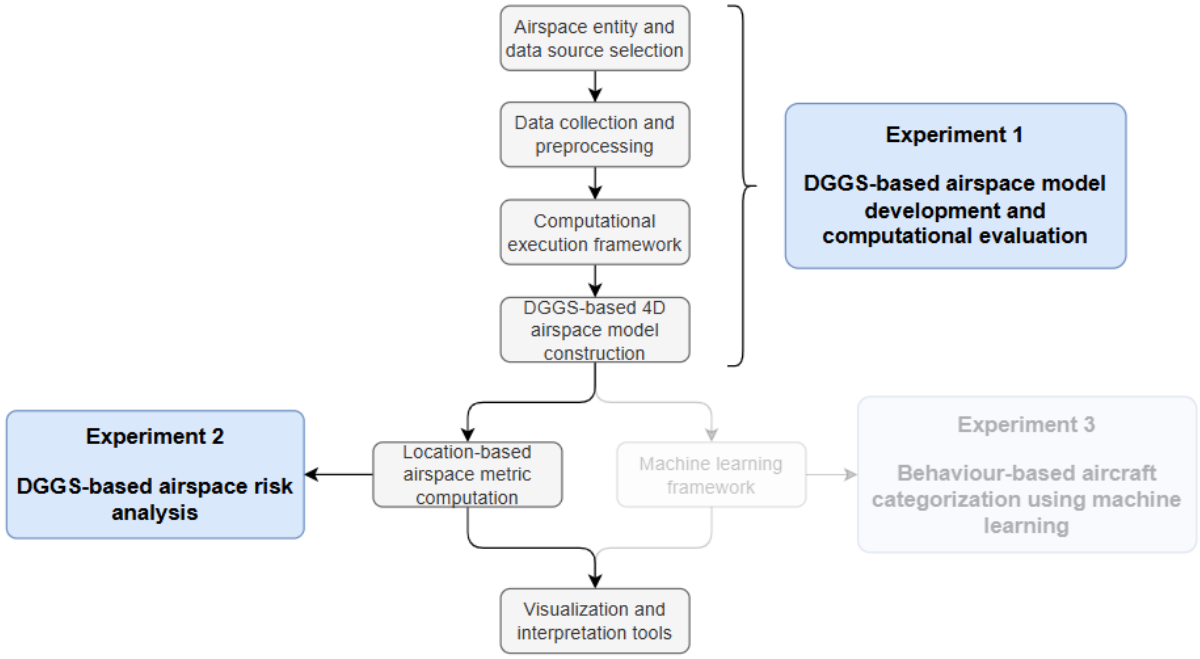


Figure 52: Relationship between methodology and Experiment #2

The outputs are examined using time-series plots, three-dimensional risk maps, and two-dimensional risk maps at different altitudes. These visualizations are used to interpret how the derived quantities vary with location, altitude, time, and the underlying aircraft and weather observations; they are not treated as separate analytical contributions.

The chapter is organized as follows. Section 6.1 defines the experimental design, inputs, outputs, and validation logic. Section 6.2 presents the risk model requirements and the mathematical formulation of the three risk indicators. Section 6.3 defines the weather-only, aircraft-only, and combined scenarios. Section 6.4 presents and interprets the resulting location-based risk fields. Section 6.5 discusses the results with respect to the broader analytical product objective of Research Question #2 and the risk-specific proposal of Hypothesis #2.

Appendix B presents the supporting review of MAC and weather risk literature that informed the risk case study. The corresponding **risk model requirements**, informed by this literature and the broader airspace-modelling literature review established in Chapter 2, are defined in Section 6.2.1. They are therefore used in this chapter to assess whether important characteristics of the underlying location-based airspace model remain available when its contents are transformed into derived risk metrics.

6.1 Experimental design, inputs, and outputs

Experiment #2 tests whether heterogeneous observations contained within the reusable DGGS-based 4D airspace model can be transformed into distinct, interpretable, location-based analytical products without redefining the underlying spatial and temporal representation. Risk analysis is used as the case study because it requires metrics derived from two physically distinct and independently processed data sources—aircraft trajectories and weather observations—to be evaluated at common three-dimensional locations and times.

Experiment #2 uses Airspace model #1 from Chapter 5. This model represents a 5 km × 5 km × 5 km terminal-area volume centred on the LGA airport, with a uniform DGGS cell width of 100 m. The resulting grid contains 125,000 three-dimensional DGGS cells. The relatively fine resolution was chosen to support detailed examination of local variations in weather, aircraft convergence, and combined risk within a compact terminal-area domain.

Airspace model #2 is not used to calculate the risk results presented in this chapter. Its larger regional domain and mixed-resolution structure were used in Experiment #1 to demonstrate airspace-model construction at a different spatial scale. Experiment #2 instead uses the finer Airspace model #1 so that the spatial and vertical behaviour of the derived risk indicators can be examined at individual DGGS locations while keeping the resulting visualizations interpretable.

Three related analytical products are derived. The first is a **weather risk metric** calculated from the positions and motions of radar-derived storm cell centroids relative to DGGS locations. The second is a **location-based MAC risk metric** calculated from the convergence of aircraft pairs toward common DGGS locations. The third is a **combined risk metric** calculated from the independently derived weather- and MAC risk metrics at common DGGS locations and time steps. The three products therefore provide a controlled progression from a metric derived from weather observations, to a second metric derived from aircraft observations, and finally to an analytical product that combines information from both sources.

The experiment is evaluated against the four **evaluation criteria** established in Section 3.2: i) the two individual risk metrics must be computable at common DGGS locations and times; ii) their spatial and temporal variation must be interpretable relative to the underlying observations; iii) the two metrics must

be combinable within the same location-based analytical field; and iv) the resulting products must be interpretable using complementary temporal, three-dimensional, and altitude-dependent representations. Table 21 summarizes the experiment inputs, outputs, and corresponding validation purpose.

Table 21: Inputs, outputs, and validation purpose of Experiment #2 components

Risk metric	Inputs	Outputs	Validation purpose
Weather risk	Airspace model #1 DGGS locations and time steps; radar-derived storm cell data; weather risk time threshold	Location-based weather risk indicator; time series; 2D and 3D weather risk fields	Demonstrate that weather observations contained within the airspace model can be transformed into a time-varying risk metric referenced to fixed 3D DGGS locations
Location-based MAC risk	Airspace model #1 DGGS locations and time steps; ADS-B aircraft trajectories; aircraft-pair convergence threshold	Location-based MAC risk indicator; valid aircraft-pair information; time series; 2D and 3D MAC risk fields	Demonstrate that a second risk metric can be derived from aircraft observations using the same spatial and temporal reference
Combined risk	Co-indexed weather- and MAC risk outputs at common DGGS locations and time steps	Combined weather–MAC risk indicator; time series; 2D and 3D combined risk fields	Demonstrate that independently derived quantities from heterogeneous sources can be combined within a common location-based analytical field

6.2 Risk metric formulation

This section defines the requirements, conceptual structure, and mathematical formulation of the location-based risk model that serves as the analytical case study for Experiment #2.

6.2.1 Motivation and risk model requirements

Experiment #2 requires a risk formulation that can be applied to the contents of the reusable airspace model developed in Experiment #1 while retaining the modelling characteristics that are relevant to downstream analysis. The objective is therefore not simply to select an existing weather risk model and an existing collision risk model. Rather, the risk case study must test whether heterogeneous aircraft and weather observations contained within the common airspace representation can be transformed into separate and combined metrics without discarding the spatial, temporal, and multi-source structure established by the base model of Experiment #1.

This requirement follows directly from the validation performed in Experiment #1. Section 5.3 and Table 20 demonstrate that the developed airspace model can integrate aircraft and weather data, retain explicit three-dimensional and temporal information, operate at different spatial extents and resolutions, and support DGGS cell-level aggregation. Section 5.4 separately evaluates the computational behaviour and reuse of the resulting model outputs. Experiment #2 therefore asks a subsequent question: *when new analytical products are derived from the model, do the modelling characteristics required for the intended application remain available in those products?*

This distinction is important because not every risk model would provide an appropriate test of the methodology developed in this thesis. For example, a risk formulation that reduces the airspace to two dimensions would discard the explicit altitude representation demonstrated in Experiment #1; a formulation tied to one fixed route or sector geometry would not retain the geographical adaptability of the location-based approach; and separate weather- and aircraft risk models that cannot be referenced to common locations and times would not demonstrate heterogeneous analytical integration. The requirements for Experiment #2 are therefore informed jointly by the broader airspace-modelling characteristics identified in Chapter 2 and demonstrated in Experiment #1, and by the risk-specific literature reviewed in Appendix B.

On this basis, the Experiment #2 risk model is designed to satisfy the following **six risk model requirements**:

1. **Air-vehicle agnosticism:** the location-based formulation should not depend on a predefined aircraft category, aerodynamic model, or vehicle-specific operational rule in order to calculate the MAC indicator from observed trajectory states.
2. **Applicability to different airspace regions:** the analytical formulation should reference DGGS locations rather than a specific route, sector, or airport geometry so that the same calculation can in principle be applied when the underlying model domain is changed.
3. **Three-dimensional representation:** the analytical quantities should retain altitude as an explicit spatial dimension rather than reducing risk to a two-dimensional horizontal surface.
4. **Time-varying analysis:** the analytical quantities should be evaluated at explicit time steps so that changes in aircraft and weather conditions are reflected in the resulting risk fields.
5. **Scalability in space, time, and entity count:** the same analytical logic should remain applicable as additional DGGS locations, time steps, or relevant airspace entities are considered, subject to the computational limits of the implementation.
6. **Multi-hazard integration**: independently derived risk metrics representing different hazard sources should be referenced to the same spatial and temporal framework so that they can subsequently be combined.

These requirements guide the development of the risk model. Experiment #2 evaluates its implementation using Airspace model #1, while applicability to other geographic regions and computational scalability beyond the selected model configuration remain outside the scope of the risk case study.

The reviewed MAC risk literature includes deterministic models such as the Reich model and related separation-minima formulations, as well as probabilistic, Monte Carlo, agent-based, and Bayesian-network approaches [467]–[479]. These approaches provide important foundations for collision risk analysis but are often tied to specific route structures, separation criteria, aircraft performance assumptions, operational procedures, or scenario-specific probability models. No single reviewed approach was therefore adopted directly as the analytical structure for the risk model required for Experiment #2.

The reviewed weather risk literature includes high-resolution weather-hazard quantification, probabilistic and ensemble forecasting, NEXRAD-supported risk-awareness tools, reconstructed weather fields, windshear and radar-based hazard detection, and airport-scale severe-weather analysis [480]–[488]. These approaches demonstrate the operational importance of weather information, but they generally produce weather-specific outputs independently of aircraft-interaction risk.

This thesis therefore develops a custom location-based risk model for Experiment #2. It is not proposed as a replacement for established collision risk, conflict detection, meteorological, or operational decision-support models. Instead, it provides a controlled implementation through which the reusable DGGS-based

airspace model can be tested as a common foundation for deriving and combining heterogeneous location-based risk metrics. The six requirements therefore provide a direct link between the modelling characteristics established in Experiment #1 and the derived analytical products evaluated in Experiment #2.

6.2.2 Conceptual description of the developed risk model

Purpose. The purpose of the developed risk model is to transform heterogeneous aircraft and weather data within the DGGS-based 4D airspace model into separate and combined analytical quantities referenced to fixed three-dimensional locations and time steps. Risk is therefore treated as a derived property of a location and time rather than as a property assigned only to an aircraft trajectory, aircraft pair, storm track, or predefined airspace region.

Concept. The developed spatiotemporal airspace risk model uses the processed aircraft trajectories, storm cell data, DGGS grid, and time references of Airspace model #1 as input and calculates risk metrics for each centroid and time step. The risk model is configurable for different regions and resolutions and is vehicle-agnostic. At each centroid c_j and time t_i in the DGGS grid, the model computes the three risk metrics described below, derived from the airspace model's entities (storm cells, aircraft) without assuming future intent or flight plans.

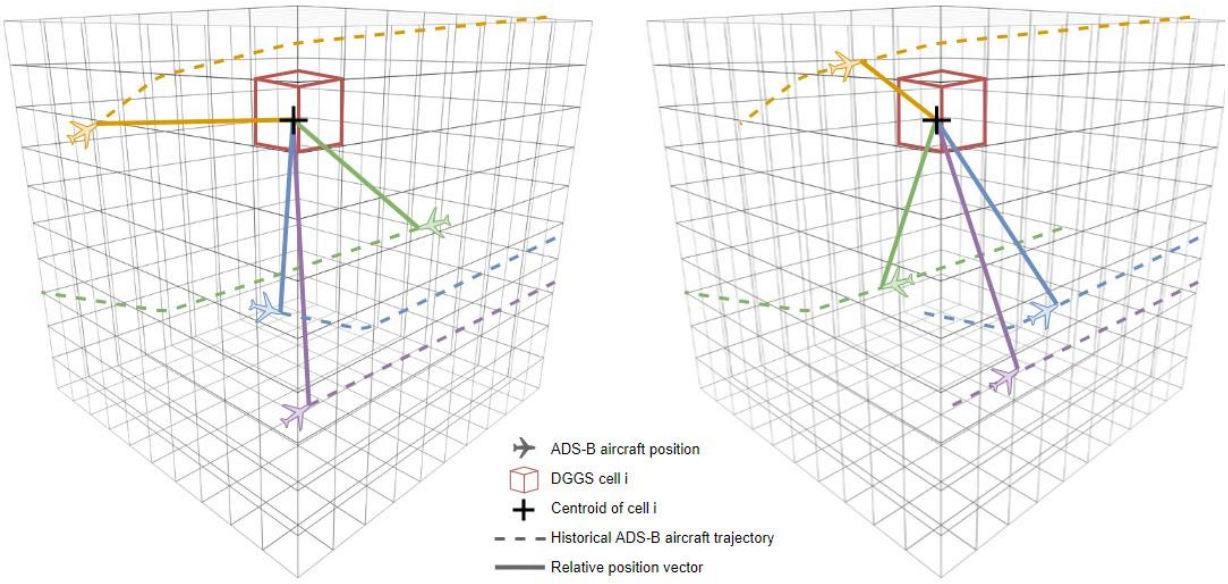
- **Weather risk.** The weather risk metric evaluates the motion of radar-derived storm cell centroids relative to each DGGS centroid. The principal variable is the estimated convergence time of a storm cell toward the airspace location being evaluated. This estimate uses a **constant-convergence-rate assumption**, meaning that the current rate at which the storm cell is approaching the DGGS centroid is assumed to remain constant when estimating the time required to reach that location. Shorter positive convergence times therefore produce higher normalized weather risk indicators.
- **Location-based MAC risk**. The MAC risk metric evaluates whether two aircraft are simultaneously converging toward the same DGGS centroid with sufficiently similar estimated arrival times. When this condition is met, the aircraft pair with the shortest relevant convergence time determines the local MAC risk value. A higher normalized MAC risk indicator therefore represents an airsapce geometry in which a qualifying aircraft pair is projected to converge toward the same three-dimensional location sooner.
- **Combined risk.** The normalized weather and MAC risk metrics are associated with the same DGGS centroid and time step, and are then combined into a single analytical field. The combined output therefore preserves the individual location and time reference of both metrics while identifying locations where weather-related and aircraft-related hazards occur simultaneously.

6.2.3 Mathematical formulation of the risk metrics

The risk model computes a metric for selected DGGS centroids at each time step. The metric evaluates the level of risk associated with a specific location in the airspace grid, regardless of whether the source of the risk is aircraft proximity, hazardous weather, or both. This location-based formulation is different from a trajectory-only formulation because the output is attached to fixed airspace locations rather than only to moving aircraft or storm cell trajectories.

The general calculation begins by computing the *shortest distance* between each DGGS centroid and each relevant airspace entity at consecutive time steps. Typical distances are illustrated using air traffic airspace entities for two different times in Figure 53a and Figure 53b for a conceptual DGGS grid. The *change in this distance* is then used to estimate whether the entity is converging toward the centroid. If the entity is converging, a *convergence time* is calculated. The convergence time represents the hypothetical time each

airspace entity (weather storm cells or ADS-B aircraft) would take to reach the centroid if it followed the shortest path at the calculated rate of change. The convergence time is therefore a local kinematic estimate based on the **constant-convergence-rate assumption**, rather than a prediction that the entity will actually pass through the evaluated centroid. A shorter positive convergence time indicates a more immediate approach to the evaluated location and consequently produces a higher risk indicator. This formulation intentionally avoids assuming that an aircraft or storm cell will continue along its complete historical trajectory or follow a known future path. The resulting quantity describes the local geometry present at the evaluated time and location, allowing for the possibility that airspace entities may not follow projected or anticipated flight paths and is indicative of the potential risk at that moment in time.



a) Shortest distance at time t1 b) Shortest distance at time t2

Figure 53: Illustration of the shortest distances at two different times

For visualization and comparison, the convergence time metrics are transformed to a normalized 0–100% worst-case-event scale using the thresholds defined below. A value approaching 100% corresponds to the limiting case in which the relevant entity or aircraft pair is at, or is predicted by the local convergence calculation to reach, the evaluated location immediately. A value approaching 0% corresponds to the selected maximum convergence time threshold or to the absence of a qualifying convergence condition. The percentage is therefore a **normalized risk indicator**, not an empirically calibrated probability of a meteorological encounter, loss of separation, or midair collision. Throughout this chapter, **risk metric** refers to the underlying application-specific calculation, whereas **normalized risk indicator** refers to the corresponding 0–100% value used for visualization and comparison. These terms are used consistently to distinguish the calculated metric from its normalized graphical representation.

6.2.3.1 Weather risk metric

The weather risk metric evaluates storm cell motion relative to each DGGS centroid. At each time step, the model identifies storm cells whose distance from the evaluated centroid is decreasing and calculates their convergence times. If more than one storm cell is converging toward the same centroid, the smallest positive

convergence time determines the weather risk metric because it represents the storm cell estimated to reach that location first under the constant-convergence-rate assumption.

A weather risk metric is calculated for each centroid in the DGGS grid at each time interval. The weather risk metric at centroid c_j and time t_i, WR_{c_j,t_i}, equals the minimum convergence time across all weather entities, as described by Equation (15). The weather risk metric WR_{c_j,t_i} quantifies, in seconds, how close the nearest storm cell is to each centroid in the DGGS grid at any given time. Smaller values indicate that storm cells converge more quickly toward the evaluated three-dimensional location.

$$WR_{c_j,t_i} = \min(CT_{s_k,c_j,t_i}, CT_{s_{k+1},c_j,t_i}, CT_{s_{k+2},c_j,t_i}, \dots) \tag{15}$$

The change in storm cell distance is defined as $\Delta d_{s_k,c_j,t_i} = d_{s_k,c_j,t_{i-1}} - d_{s_k,c_j,t_i}$, such that a positive value indicates that storm cell s_k is converging toward centroid c_j. For a converging storm cell, $v_{s_k,c_j,t_i} = \Delta d_{s_k,c_j,t_i}/\Delta t$ is its convergence velocity and $CT_{s_k,c_j,t_i} = d_{s_k,c_j,t_i}/v_{s_k,c_j,t_i}$ is its estimated convergence time.

In Equation (15), $WR_{c_j,t_i}$is the weather risk metric at DGGS centroid c_j and analysis time t_i; s_k is storm cell k; and CT_{s_k,c_j,t_i} is the convergence time of storm cell $s_k$toward centroid $c_j$at t_i. Only storm cells having a decreasing centroid distance and a valid positive convergence time are included in the minimum. Consequently, WR_{c_j,t_i}represents the shortest qualifying convergence time among the storm cells evaluated for that centroid and time.

For comparison and visualization, the weather convergence time is mapped to a normalized worst-case-event risk indicator scale. The upper limiting condition occurs when a storm cell centroid coincides with the evaluated DGGS centroid, corresponding to a convergence time of 0 s and the maximum normalized indicator. The lower end of the displayed scale is defined using a **1,800 s (30 min) weather convergence time threshold**. Storm cells with progressively shorter qualifying convergence times therefore produce progressively larger indicator values.

The 1,800 s threshold is used as an application-specific scaling parameter for this case study because the relevant storm cell motion evolves over a substantially longer time scale than the aircraft interactions examined below. It should not be interpreted as a meteorological warning threshold or as a universal operational criterion. Different applications may select a different time horizon while retaining the same location-based formulation.

Figure 54 presents the pseudocode summarizing the implemented weather risk calculation in *risk_model.py*, and Table 22 defines the variables used by the Experiment #2 risk formulation.

```
FOR time t_i in time interval:
    FOR centroid c_j in geospatial grid:
        FOR storm cell s_k in airspace region of interest:
            Calculate the shortest distance d_{s_k,c_j,t_i}
            Calculate the rate of change of the shortest distance Δd_{s_k,c_j,t_i} = d_{s_k,c_j,t_{i-1}} − d_{s_k,c_j,t_i}
            IF storm cell s_k is converging to centroid c_j (Δd_{s_k,c_j,t_i} > 0):
                Calculate velocity v_{s_k,c_j,t_i} = Δd_{s_k,c_j,t_i}/Δt
                Calculate convergence time CT_{s_k,c_j,t_i} = d_{s_k,c_j,t_i}/ v_{s_k,c_j,t_i}
    Calculate the weather risk for centroid c_j at time t_i as WR_{c_j,t_i} = min(CT_{s_k,c_j,t_i}, CT_{s_{k+1},c_j,t_i}, CT_{s_{k+2},c_j,t_i}, ... , CT_{s_n,c_j,t_i})
```

Figure 54: Pseudocode used to calculate the weather risk metric

Table 22: List of variables used in Experiment #2 and their descriptions

Variable	Variable Type	Description
t_i	Common variables	Current time
t_{i-1}		Previous time
Δt		Time interval between t_i and t_{i-1}
c_j		Centroid i in the geospatial grid
s_k	Weather risk-specific variables	Storm cell k in the airspace region of interest
d_{s_k,c_j,t_i}		Shortest distance between storm cell s_k and centroid c_j at time t_i
$d_{s_k,c_j,t_{i-1}}$		Shortest distance between storm cell s_k and centroid c_j at time t_{i-1}
$\Delta d_{s_k,c_j,t_i}$		The rate of change of the shortest distance between d_{s_k,c_j,t_i} and $d_{s_k,c_j,t_{i-1}}$
v_{s_k,c_j,t_i}		Velocity of storm cell s_k converging toward centroid c_j at time t_i
CT_{s_k,c_j,t_i}		Convergence time of storm cell s_k converging toward centroid c_j at time t_i (i.e., the time for storm cell s_k to arrive at centroid c_j using velocity v_{s_k,c_j,t_i})
WR_{c_j,t_i}		Weather risk for centroid c_j at time t_i. Corresponds to the minimum out of all the CT_{s_k,c_j,t_i} for each storm cell k at time t_i
a_k	Air traffic risk-specific variables	Aircraft k in the airspace region of interest
d_{a_k,c_j,t_i}		Shortest distance between aircraft a_k and centroid c_j at time t_i
$d_{a_k,c_j,t_{i-1}}$		Shortest distance between aircraft a_k and centroid c_j at time t_{i-1}
$\Delta d_{a_k,c_j,t_i}$		The rate of change of the shortest distance between d_{a_k,c_j,t_i} and $d_{a_k,c_j,t_{i-1}}$
v_{a_k,c_j,t_i}		Velocity of aircraft a_k converging toward centroid c_j at time t_i
CT_{a_k,c_j,t_i}		Convergence time of aircraft a_k converging toward centroid c_j at time t_i (i.e., the time for aircraft a_k to arrive at centroid c_j using velocity v_{a_k,c_j,t_i})
AP_m		Aircraft pair m out of all possible unique aircraft pairs present in the airspace
$\Delta CT_{AP_m,c_j,t_i}$		The difference between convergence times for pair of aircraft AP_m
$\Delta CT_{threshold}$		Arbitrary threshold used for filtering $\Delta CT_{AP_m,c_j,t_i}$ values based on desired granularity (use case specific)
CT_{AP_{m1},c_j,t_i}		Convergence time of the 1st aircraft in the aircraft pair AP_m
CT_{AP_{m2},c_j,t_i}		Convergence time of the 2nd aircraft in the aircraft pair AP_m
CT_{min,AP_m,c_j,t_i}		Minimum convergence time of the aircraft pair AP_m
$MACR_{AP_m,c_j,t_i}$		Air traffic risk of the aircraft pair AP_m for centroid c_j at time t_i
$MACR_{c_j,t_i}$		Air traffic risk for centroid c_j at time t_i. Corresponds to the minimum out of all the $MACR_{AP_m,c_j,t_i}$
$WR^N_{c_j,t_i}$	Combined risk-specific variables	Normalized weather risk indicator at centroid c_j and time t_i; dimensionless and bounded between 0 and 1.
$MACR^N_{c_j,t_i}$		Normalized location-based MAC risk indicator at centroid c_j and time t_i; dimensionless and bounded between 0 and 1.
CR_{c_j,t_i}		Combined weather–MAC risk indicator at centroid c_j and time t_i; dimensionless and bounded between 0 and 1.

6.2.3.2 Location-based mid-air collision (MAC) risk metric

The location-based MAC risk metric, $MACR_{c_j,t_i}$, evaluates whether pairs of aircraft are converging toward the same DGGS centroid within sufficiently similar time intervals. The resulting quantity is intended to identify three-dimensional locations where the instantaneous motion of two aircraft is consistent with a potentially hazardous convergence geometry.

A candidate aircraft pair contributes to the MAC risk indicator at centroid c_j and time t_i when two conditions are met. First, both aircraft must be converging toward the evaluated centroid, and the difference between their convergence times must fall within the specified pairwise threshold $\Delta CT_{\text{threshold}}$. This condition identifies aircraft that, under their instantaneous motion relative to the centroid, would arrive at approximately the same location at approximately the same time. The threshold $\Delta CT_{\text{threshold}}$ can be adjusted to provide a margin around this temporal coincidence, such that aircraft having sufficiently similar convergence times form a qualifying candidate pair. Second, when multiple aircraft pairs satisfy this condition, their pair-level MAC risk metrics are compared and the pair having the smallest value is treated as the critical pair for that centroid and time.

Equation (16) defines the selection of the local MAC risk metric $MACR_{c_j,t_i}$ from the qualifying aircraft pairs.

$$MACR_{c_j,t_i} = \min(MACR_{AP_m,c_j,t_i}, MACR_{AP_{m+1},c_j,t_i}, MACR_{AP_{m+2},c_j,t_i}, \dots, MACR_{AP_n,c_j,t_i}) \quad (16)$$

Converging aircraft are first ordered according to CT_{a_k,c_j,t_i}, and aircraft pairs AP_m are formed from adjacent convergence-time values. A pair qualifies when $\Delta CT_{AP_m,c_j,t_i}$ is smaller than the selected $\Delta CT_{\text{threshold}}$. Among the qualifying pairs, the pair having the smallest $MACR_{AP_m,c_j,t_i}$ determines the location-based MAC risk metric $MACR_{c_j,t_i}$.

The pair-level MAC risk metric $MACR_{AP_m,c_j,t_i}$ is constructed from two quantities: the difference between the two aircraft convergence times and the smaller of those convergence times. Equations (17) and (18) define these quantities, respectively, and Equation (19) combines them to define the pair-level MAC risk metric.

$$\Delta CT_{AP_m,c_j,t_i} = \left| CT_{AP_{m1},c_j,t_i} - CT_{AP_{m2},c_j,t_i} \right| \quad (17)$$

$$CT_{\min,AP_m,c_j,t_i} = \min(CT_{AP_{m1},c_j,t_i}, CT_{AP_{m2},c_j,t_i}) \quad (18)$$

$$MACR_{AP_m,c_j,t_i} = \Delta CT_{AP_m,c_j,t_i} + CT_{min,AP_m,c_j,t_i} \quad (19)$$

In Equations (17)–(19), CT_{AP_{m1},c_j,t_i} and CT_{AP_{m2},c_j,t_i} are the convergence times of the first and second aircraft in qualifying pair AP_m, respectively. $\Delta CT_{AP_m,c_j,t_i}$ is the absolute difference between these convergence times, while $CT_{\min,AP_m,c_j,t_i}$ is the smaller of the two. The pair-level metric $MACR_{AP_m,c_j,t_i}$ therefore accounts for both the similarity of the estimated arrival times at the evaluated centroid and the immediacy of that potential convergence. Lower values correspond to aircraft pairs whose estimated convergence times are both more closely matched and more immediate.

The physical interpretation of the MAC risk metric is important. A higher normalized MAC risk indicator does not imply that the numerical probability of an actual midair collision has been estimated to be higher.

Rather, it indicates that the observed aircraft geometry is more consistent with a potentially hazardous local convergence: two aircraft are approaching the same three-dimensional DGGS location, their estimated arrival times at that location are more similar, and the projected convergence occurs within a shorter time interval. Conversely, a lower value indicates that the qualifying convergence is less immediate or that the aircraft are less closely matched in their estimated arrival times.

For the MAC risk metric, the convergence-time result is mapped to a normalized worst-case-event risk indicator scale for ease of interpretation and visualization. The upper-limiting case occurs when a qualifying aircraft pair is located at the evaluated centroid at the same time. A **convergence-time threshold of 180 s (3 min)** defines the lower end of the displayed MAC risk scale for the case study. The shorter threshold relative to the weather risk metric reflects the substantially faster motion and shorter interaction time scale of aircraft.

The 180 s value is an application-specific analytical threshold. It is not a prescribed separation minimum, a certification criterion, or an operational collision-warning threshold. Changing this value would alter the indicator's temporal sensitivity without affecting its underlying location-based formulation.

Figure 55 presents the pseudocode summarizing the implemented MAC risk calculation in *risk_model.py*

FOR time t_i in time interval:
  FOR centroid c_j in geospatial grid:
    FOR aircraft a_k in airspace region of interest:
      Calculate the shortest distance d_{a_k,c_j,t_i}
      Calculate the rate of change of the shortest distance $\Delta d_{a_k,c_j,t_i} = d_{a_k,c_j,t_{i-1}} - d_{a_k,c_j,t_i}$
      IF aircraft a_k is converging to centroid c_j ($\Delta d_{a_k,c_j,t_i} > 0$):
        Calculate velocity $v_{a_k,c_j,t_i} = \Delta d_{a_k,c_j,t_i}/\Delta t$
        Calculate convergence time $CT_{a_k,c_j,t_i} = d_{a_k,c_j,t_i}/\, v_{a_k,c_j,t_i}$
    FOR aircraft pair AP_m in every pair of aircraft:
      Calculate the difference in convergence time $\Delta CT_{AP_m,c_j,t_i}$
    FOR all the $\Delta CT_{AP_m,c_j,t_i}$ where both convergence times of pair AP_m are the same within a threshold $\Delta CT_{threshold}$:
      Find the minimum convergence time of the pair as $CT_{min,AP_m,c_j,t_i} = min(CT_{AP_{m1},c_j,t_i}, CT_{AP_{m2},c_j,t_i})$
      Calculate the air traffic risk of the aircraft pair AP_m as $MACR_{AP_m,c_j,t_i} = \Delta CT_{AP_m,c_j,t_i} + CT_{min,AP_m,c_j,t_i}$
  Calculate the air traffic risk for centroid c_j at time t_i as $MACR_{c_j,t_i} = \min(MACR_{AP_m,c_j,t_i}, MACR_{AP_{m+1},c_j,t_i}, MACR_{AP_{m+2},c_j,t_i}, \dots, MACR_{AP_n,c_j,t_i})$

Figure 55: Pseudocode used to calculate the MAC risk metric

6.2.3.3 Combined risk metric

The combined risk metric integrates the normalized weather- and MAC risk indicators at the same DGGS centroid and time step. Because the two metrics are independently derived from heterogeneous source data but share a common spatial and temporal reference, they can be combined without redefining or resampling the underlying airspace representation. The combined metric is useful because real airspace operations are not affected by one hazard type at a time. Weather and traffic often affect the same airspace volume simultaneously.

Let $WR^N_{c_j,t_i}$ and $MACR^N_{c_j,t_i}$ denote the normalized weather- and MAC risk indicators, respectively. The combination uses the complementary-product form described by Ross [489], shown in Equation (20). This mathematical form is analogous to the union rule used for independent probabilities; however, the inputs in this thesis are normalized analytical risk indicators rather than empirically calibrated probabilities. Equation (20) is therefore used as a bounded aggregation rule: if either individual indicator increases, the

combined indicator increases, and when both indicators are non-zero the combined value is greater than either individual value while remaining bounded between 0 and 1.

$$CR_{c_j,t_i} = 1 - [(1 - WR_{c_j,t_i}) \times (1 - MACR_{c_j,t_i})] \quad (20)$$

In Equation (20), $WR^N_{c_j,t_i}$ is the normalized weather risk indicator, $MACR^N_{c_j,t_i}$ is the normalized MAC risk indicator, and CR_{c_j,t_i} is the resulting combined weather–MAC risk indicator at centroid c_j and time t_i. All three quantities are dimensionless and bounded between 0 and 1, with larger values representing greater indicated risk.

The use of this complementary-product formulation does not imply that hazardous weather and aircraft-interaction risk are statistically independent in operational airspace. Weather can affect aircraft routing, altitude selection, traffic concentration, and consequently aircraft-interaction geometry. The formulation is used here to provide a transparent and tractable demonstration of how two independently calculated risk metrics can be combined once they share the same DGGS location and time reference. The implications of this simplifying assumption are discussed further in Section 6.5.3.

The resulting combined value is therefore interpreted as a normalized multi-hazard analytical indicator rather than as a calibrated joint probability of an adverse operational event. In Experiment #2, its purpose is to demonstrate that separately derived quantities from heterogeneous data sources can be co-indexed and combined at common three-dimensional locations and times.

6.3 Experimental scenarios

Experiment #2 is evaluated using three scenario types with increasing analytical complexity. The weather-only scenario derives a location-based weather risk metric from the processed weather observations. The aircraft-only scenario derives a physically distinct MAC risk metric from ADS-B aircraft trajectories using the same DGGS reference structure. The combined scenario then evaluates whether the independently derived weather- and MAC risk metrics can be represented and combined at common DGGS locations and time steps.

Table 23 summarizes the different scenarios used in Experiment #2.

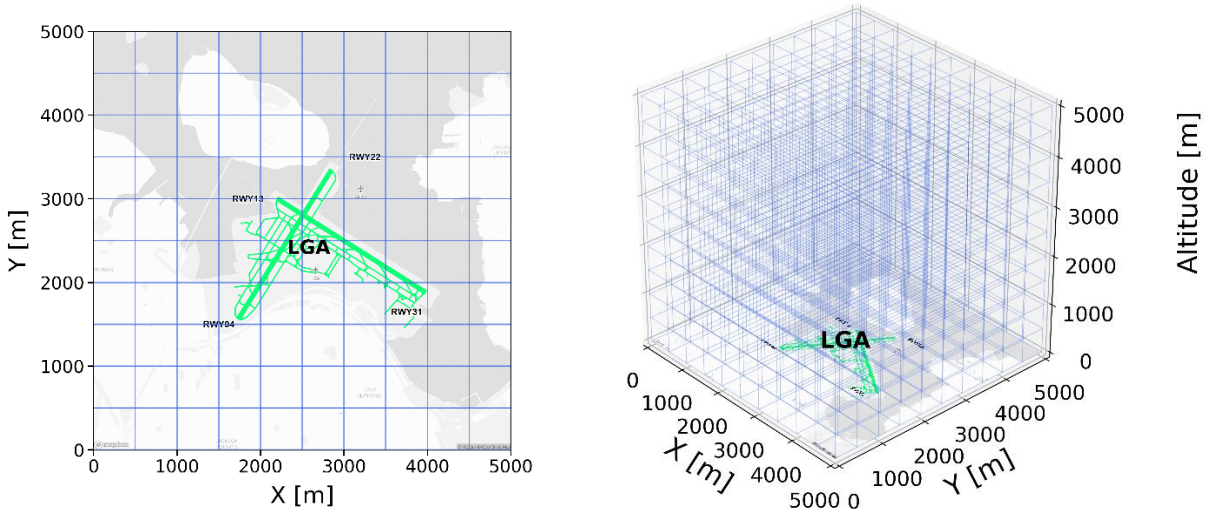
Table 23: Description of the different scenarios used in Experiment #2

Scenario type	Data used	Metric produced	Purpose
Weather-only	Radar-derived storm cell data	Weather risk indicator	Evaluate whether processed weather observations can be transformed into an interpretable, time-varying analytical field at fixed 3D DGGS locations
Aircraft-only	ADS-B aircraft trajectory data	Location-based MAC risk indicator	Evaluate whether aircraft-motion observations can be transformed into a distinct analytical field using the same DGGS locations and time reference
Combined	Storm cell and ADS-B data	Combined weather–MAC risk indicator	Evaluate whether separately derived risk metrics from heterogeneous sources can be combined at common DGGS locations and time steps.

This progression allows the behaviour of each analytical product to be interpreted relative to its source observations before evaluating their combination. It also separates the evidence required for Hypothesis #2: the first two scenarios test whether weather and aircraft-proximity indicators produce interpretable spatial and temporal patterns, whereas the third tests whether their effects can be represented jointly within the same location-based analytical field.

6.4 Location-based risk metric results

This section presents the location-based analytical results for Experiment #2. All risk calculations use Airspace model #1, the high-resolution, LGA-centred model developed in Chapter 5. Figure 56 illustrates the Airspace model #1 domain and its DGGS base grid, overlaid in blue, to provide spatial context for the risk analysis. The subsequent figures use selected DGGS centroids and time intervals to examine the spatial and temporal behaviour of the derived risk indicators.



a) 2D view of Airspace model #1

b) 3D view of Airspace model #1

Figure 56: Airspace model used to derive risk metric results

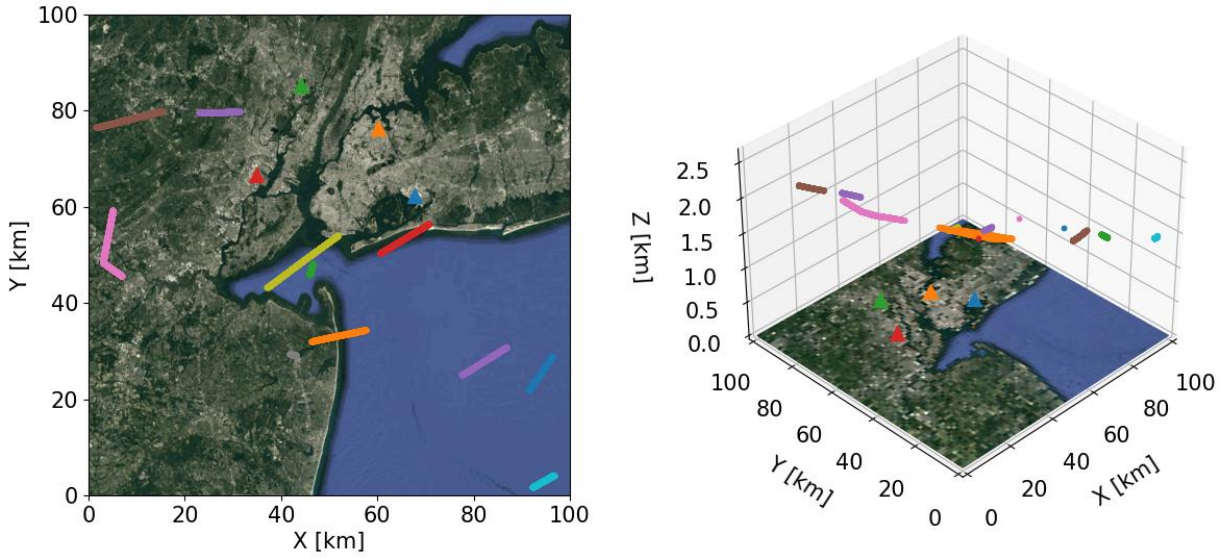
The results are presented using the three experimental scenarios. Section 6.4.1 examines the weather-derived analytical field, Section 6.4.2 examines the aircraft-derived MAC field, and Section 6.4.3 examines their combination. The figures are used to relate changes in the normalized indicators to the underlying aircraft or storm cell geometry.

6.4.1 Weather risk results

This section evaluates the first derived analytical product using radar-derived storm cell data. The weather risk indicator is evaluated at fixed DGGS locations and time steps based on the position and motion of the identified storm cell centroids.

Figure 57 provides regional weather context for the selected period using a 100 km × 100 km × 14 km view surrounding the smaller Airspace model #1 domain used for the detailed risk calculations. The two-hour regional view contains 11 identified storm cell trajectories, with horizontal displacement predominantly from southwest to northeast across the plotted region. The three-dimensional view shows the processed

storm cell centroids at different altitude levels, ranging approximately from 700 to 2,000 m MSL. These centroid altitudes represent the processed weather entities used by the analytical model and provide spatial context for the subsequent location-based risk results.

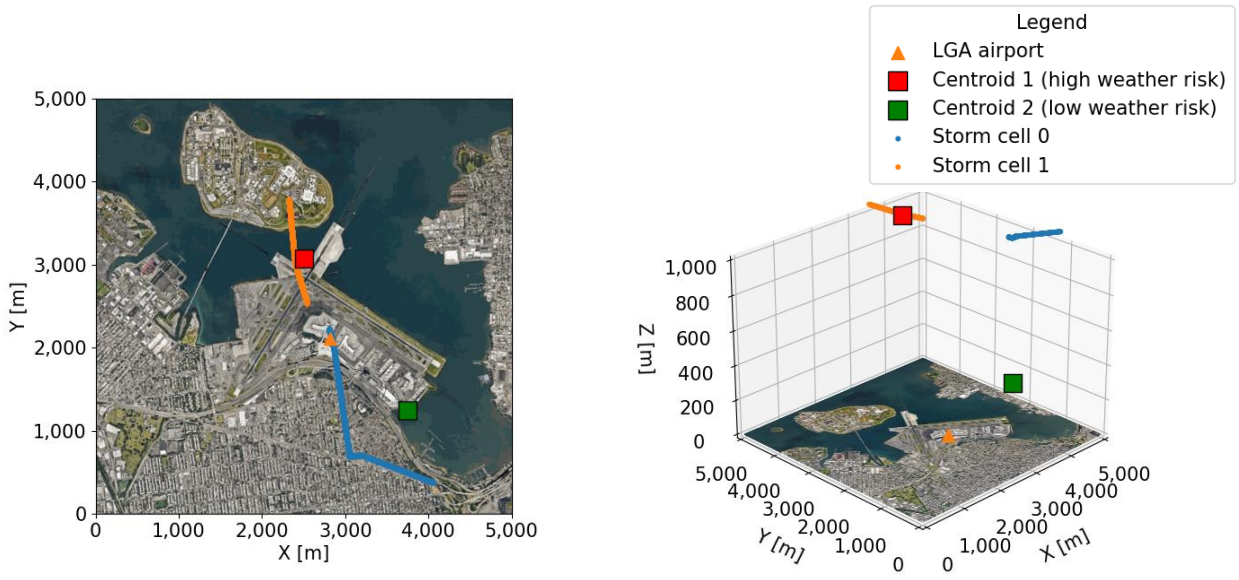


a) 2D satellite view

b) 3D satellite view

Figure 57: High-level storm cell context for the weather risk scenario

Figure 58 shows the corresponding weather entities within the 5 km × 5 km × 5 km Airspace model #1 domain. Although weather risk can be calculated for any of the 125,000 centroids in Airspace model #1, two are selected for detailed interpretation. Centroid 1 is at 1,200 m, within the altitude range occupied by the selected storm cell trajectories, whereas centroid 2 is at 300 m, below the principal altitude range occupied by the storms. Storm cells 0 and 1 are also shown so that changes in the risk indicator can be related directly to their movement.



a) 2D satellite view

b) 3D satellite view

Figure 58: 5km region near LGA airport with centroids 1 and 2 and storm data

Figure 59 shows the normalized weather risk indicators for centroids 1 and 2 during the six-minute interval when storm cells 0 and 1 affect the Airspace model #1 domain. The two locations exhibit markedly different temporal profiles. At 18:02:08, the indicator is approximately 79% at centroid 1 and then increases toward its local maximum near 18:06:00. It then decreases rapidly as the storm cell geometry relative to the centroid changes, reaching approximately 12% at 18:07:49. Centroid 2 begins at approximately 37% at 18:02:08 and decreases to 0% by approximately 18:03:45.

These values have a direct geometric interpretation under the proposed formulation. The increasing value at centroid 1 indicates that a qualifying storm cell is approaching that three-dimensional location, with a progressively shorter estimated convergence time. The decrease at centroid 2 indicates that the relevant storm cell motion no longer produces an approach toward that location. The numerical percentages should therefore be interpreted as normalized weather risk indicators.

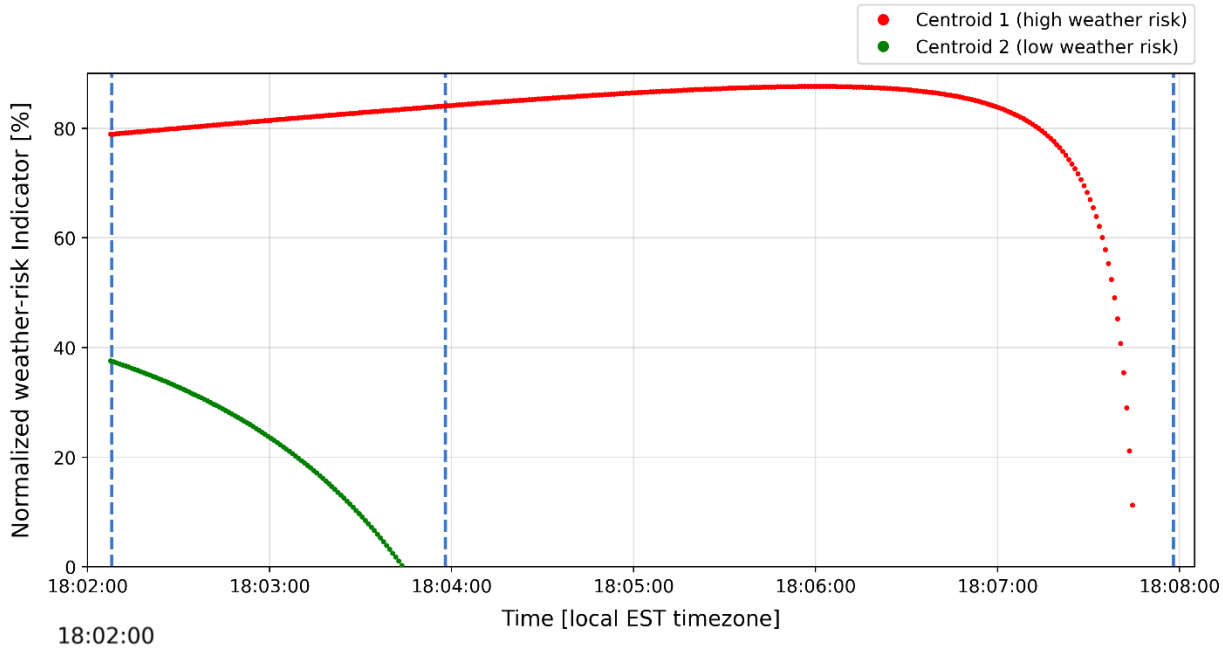


Figure 59: Weather risk metric over time for centroids 1 and 2

Figures 60-64 show the spatial distributions at 18:02:08, 18:03:58, and 18:07:58, which are marked by the vertical blue dashed lines in Figure 59. Figures 60-62 presents the complete three-dimensional weather risk indicator field at three selected times, while Figures 63-64 present horizontal slices at 1,200 m and 300 m, corresponding to the altitudes of centroids 1 and 2, respectively. The colour scale represents the normalized weather risk indicator at the evaluated DGGS locations, while the storm cell trajectories and instantaneous positions provide geometric context for interpreting the field.

Figures 60-62 should be interpreted as a location-based convergence-time field rather than as a representation of the physical storm footprint. At each DGGS location, the colour represents the normalized weather risk indicator derived from the shortest qualifying convergence time of the storm cell centroids toward that location. Consequently, the highest indicator values are not necessarily centred on the instantaneous storm cell positions; they occur at locations toward which a storm cell is currently converging with a short estimated convergence time under the constant-convergence-rate assumption. The storm cell trajectories and instantaneous positions are overlaid to provide the geometric context for this calculation.

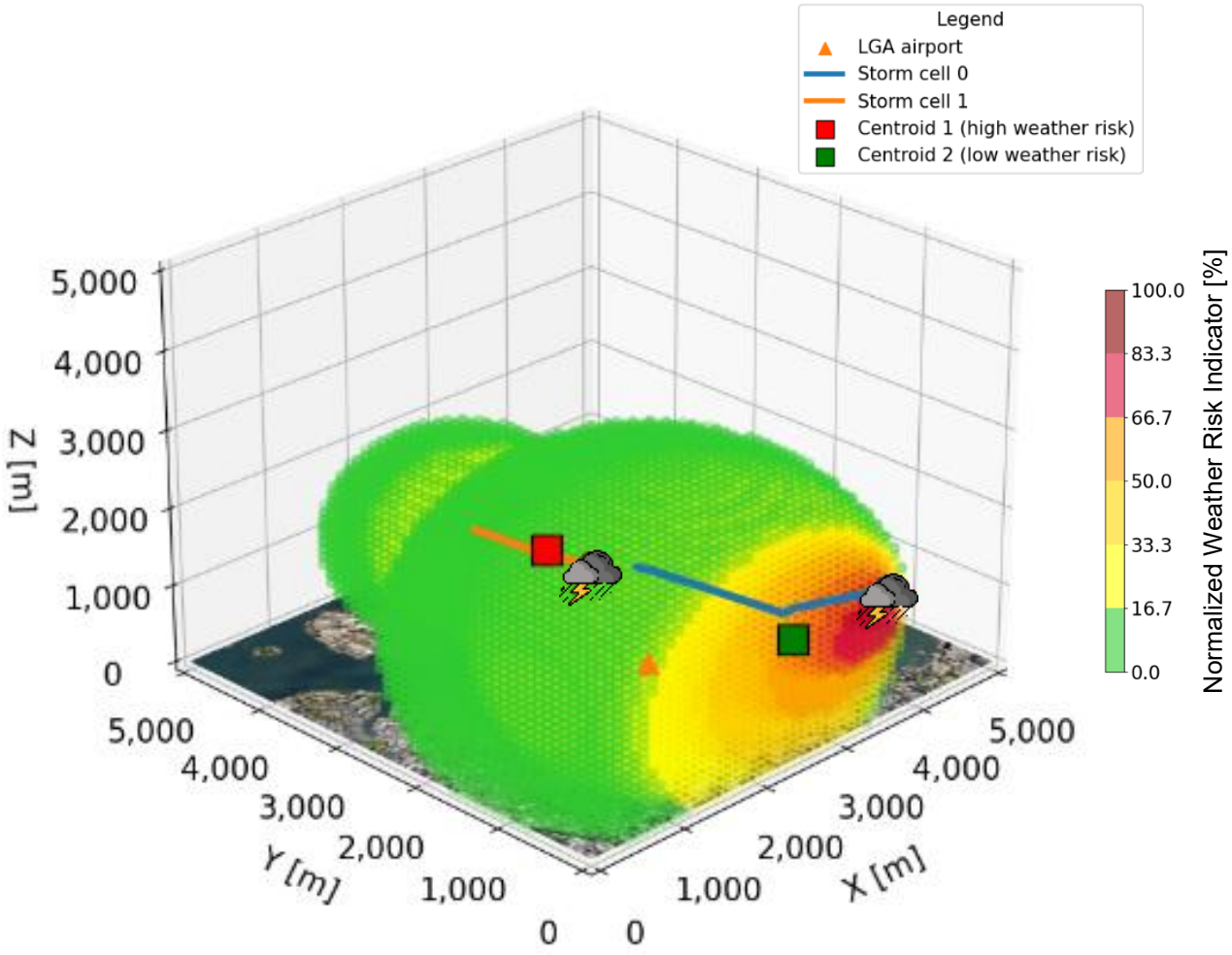


Figure 60: 3D normalized weather risk indicator field at 2022-11-22 18:02:08

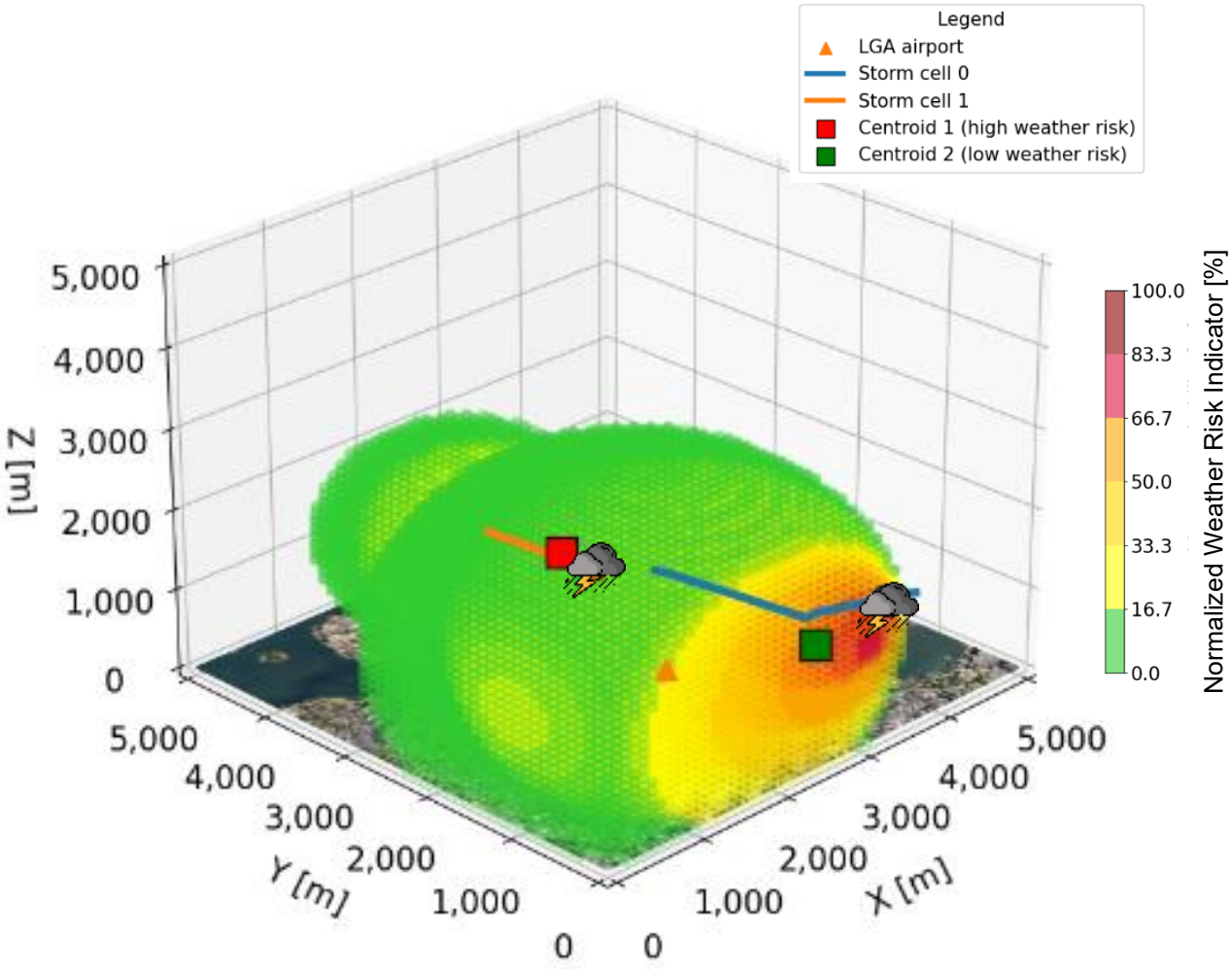


Figure 61: 3D normalized weather risk indicator field at 2022-11-22 18:03:58

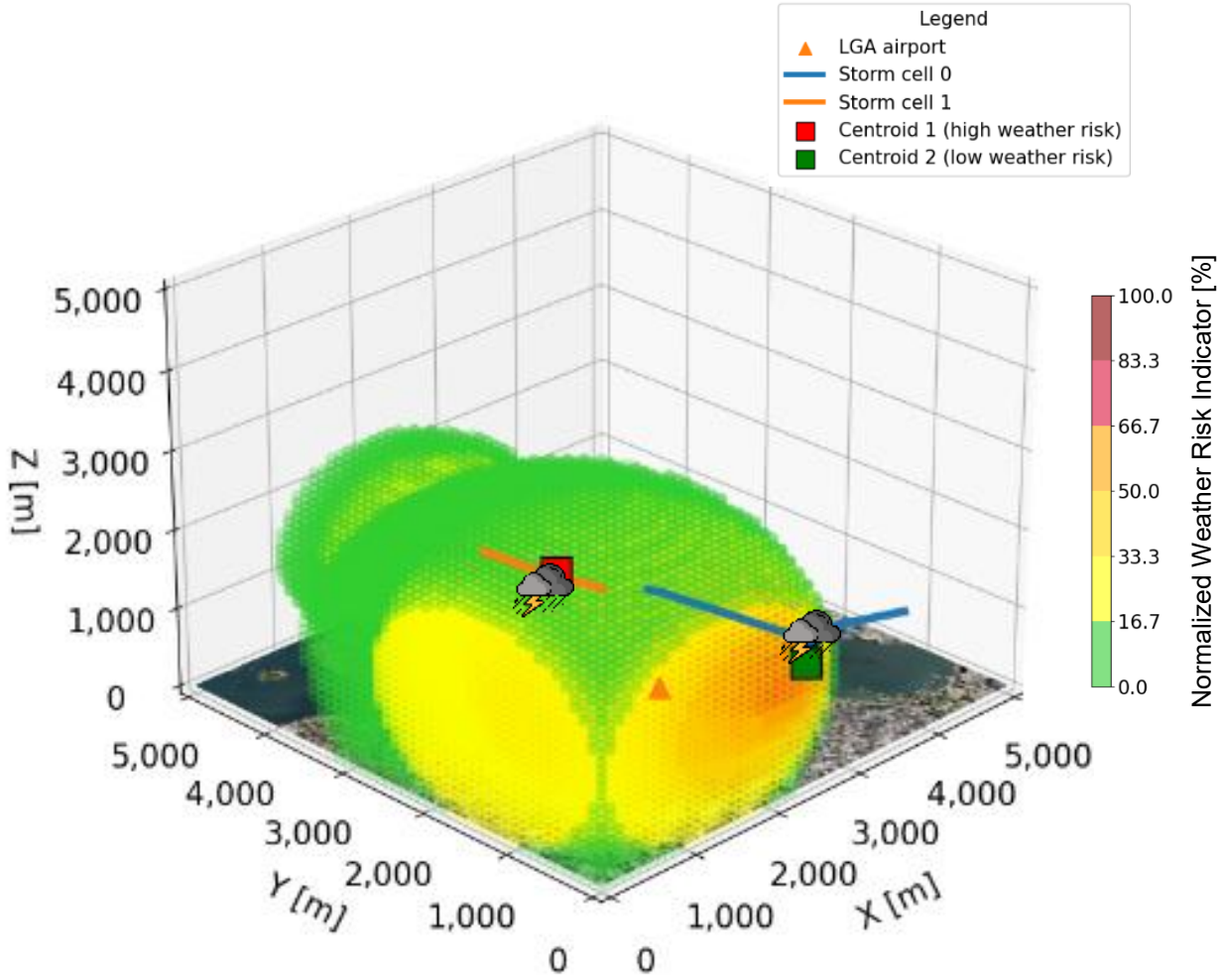


Figure 62: 3D normalized weather risk indicator field at 2022-11-22 18:07:58

Figure 63 presents the horizontal weather risk indicator field at 1,200 m, corresponding to the altitude of centroid 1. The storm cell trajectories and instantaneous positions are projected onto the horizontal plane to provide spatial context; their displayed x–y positions do not imply that the storm cell centroids occupy the 1,200 m plane. At 18:02:08, centroid 1 lies near the horizontal projection of storm cell 1's trajectory. By 18:03:58, the changing three-dimensional geometry produces a larger weather risk indicator at centroid 1. At 18:07:58, the storm cell geometry has changed again, and a different storm cell determines the local minimum convergence time.

Figure 64 presents the corresponding horizontal field at 300 m, the altitude of centroid 2. At 18:02:08, storm cell 0 produces a non-zero weather risk indicator at centroid 2. By 18:03:58, its motion no longer satisfies the convergence condition for that location and the indicator becomes 0%. Together, the two altitude slices illustrate that the resulting analytical field varies with both horizontal position and altitude as the three-dimensional storm cell geometry evolves.

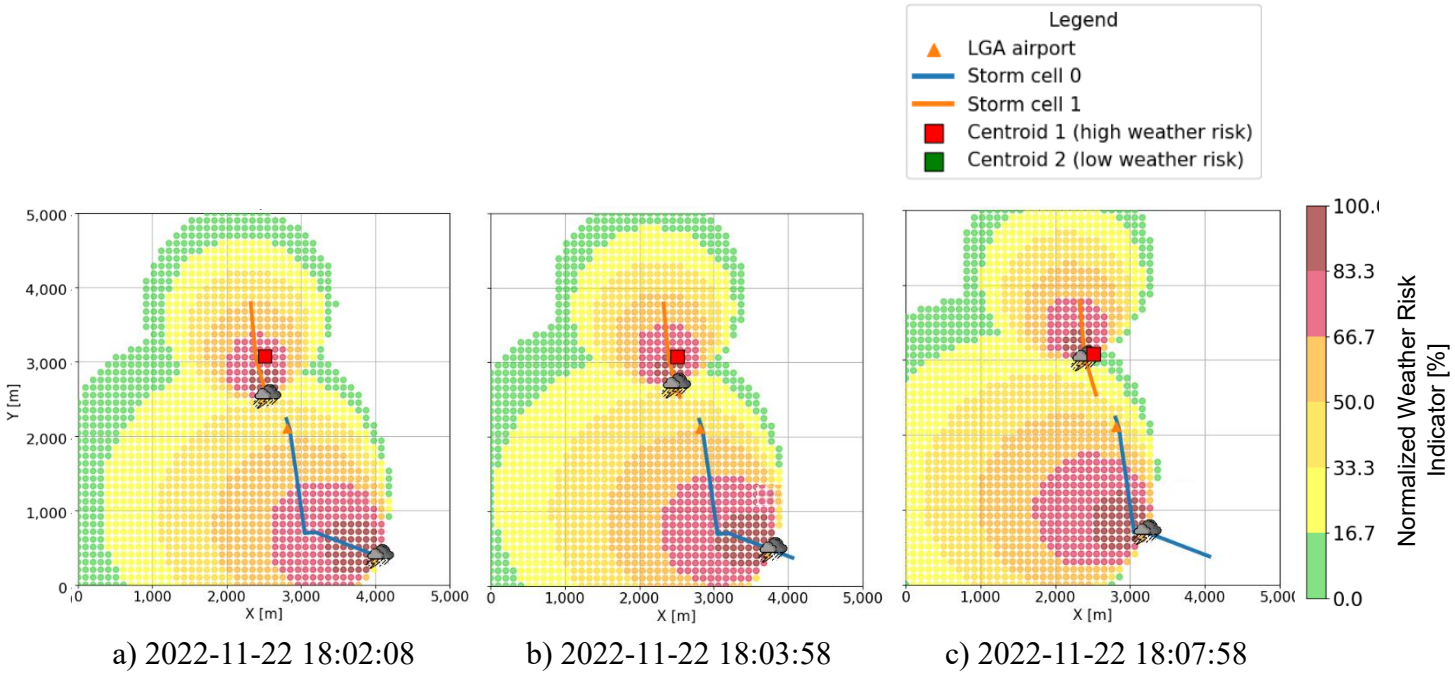


Figure 63: 2D weather risk indicator field at 1,200 m altitude for three selected times

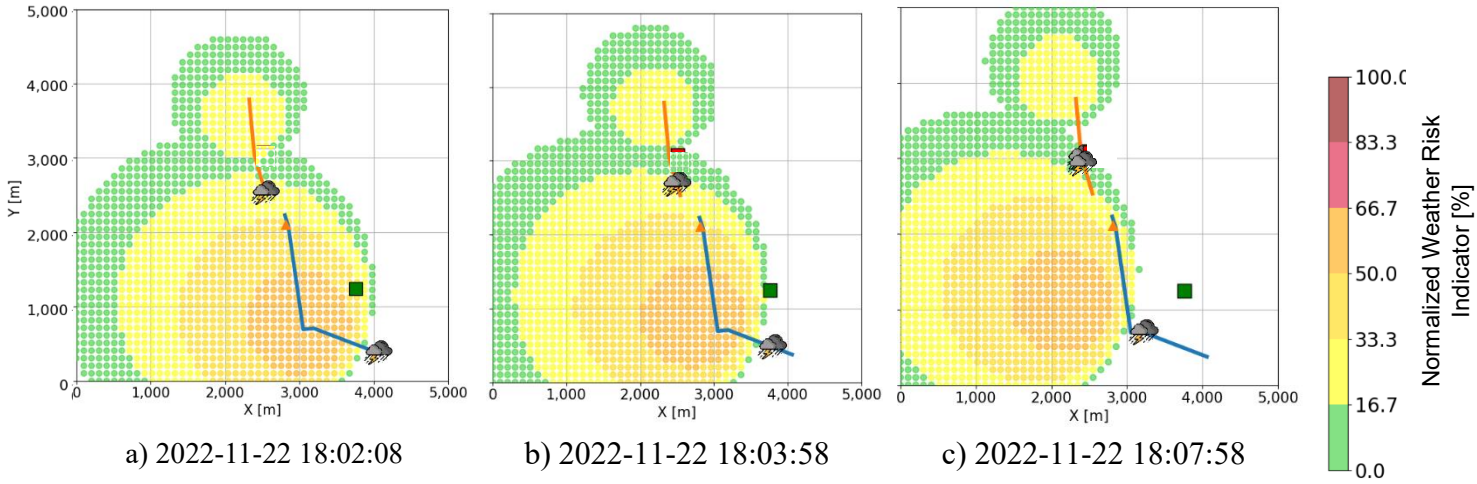


Figure 64: 2D weather risk indicator field at 300 m altitude for three selected times

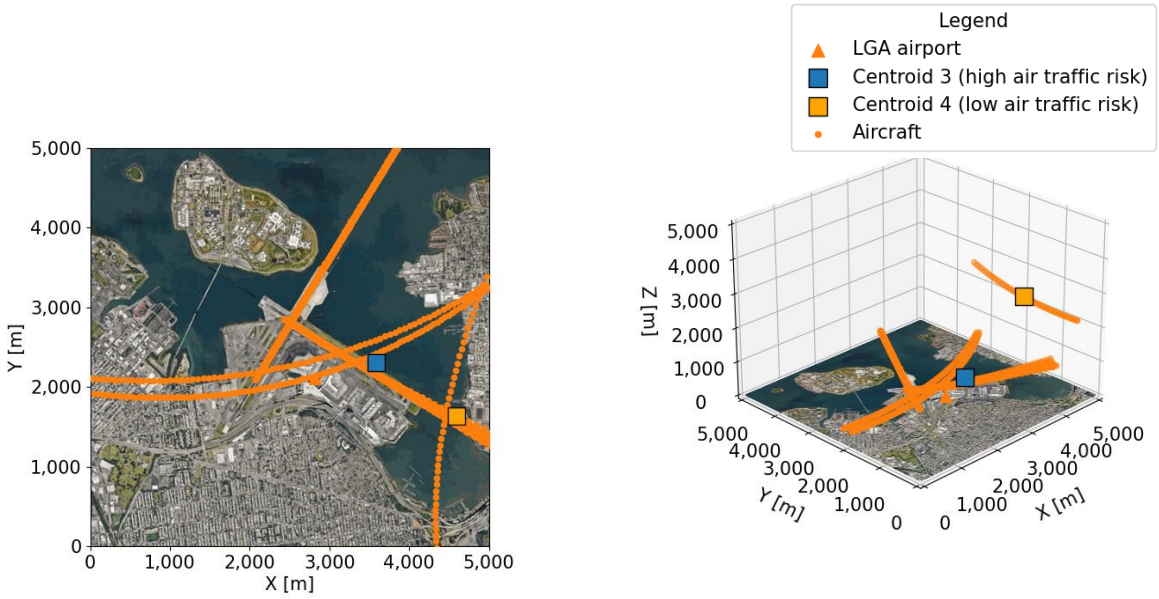
6.4.2 Location-based MAC risk results

This section evaluates the second derived analytical product using ADS-B aircraft trajectories contained within Airspace model #1. The location-based MAC risk indicator identifies DGGS locations where aircraft pairs are simultaneously converging and have sufficiently similar estimated arrival times. The objective is to determine whether aircraft interactions can be represented as an interpretable, three-dimensional, time-varying field referenced to fixed airspace locations.

The metric should be interpreted as an indicator of local aircraft-convergence geometry. A higher normalized value indicates that a qualifying aircraft pair is approaching the same three-dimensional DGGS location with similar estimated arrival times. In operational terms, the observed geometry is therefore more consistent with a potentially hazardous near-term convergence at that location.

Two scenarios are used. The first examines a selected pair of aircraft so that the spatial behaviour of a single qualifying pair can be interpreted directly. The second examines three selected aircraft, allowing multiple aircraft pairs to contribute to the field.

Figure 65 provides air-traffic context for a representative one-hour subset, from 18:00:00 to 19:00:00 EST, of the complete five-hour Airspace Model #1 interval. During this one-hour subset, 32 aircraft are represented within the model domain using overlapping orange points. The observed traffic is concentrated primarily below approximately 1,000 m, associated with arrivals and departures near LGA, with additional trajectories occupying intermediate and higher altitudes. The shorter interval is used in Figure 65 so that the individual aircraft trajectories relevant to the subsequent MAC risk examples remain visually distinguishable. Centroid 3 is located at 500 m, whereas centroid 4 is located at 2,400 m.



a) 2D satellite view

b) 3D satellite view

Figure 65: 5km region near LGA airport with centroids 3 and 4 and air traffic data

Figure 66 shows the normalized MAC risk indicators, as defined in Section 6.2.3.2, for centroids 3 and 4 over a ten-minute interval containing ten of the 32 aircraft shown in Figure 65. The secondary vertical axis identifies the aircraft present during different portions of this interval. Two cases are selected for detailed examination in the following subsections: a two-aircraft case in Section 6.4.2.1 and a three-aircraft case in Section 6.4.2.2. The blue dashed lines mark the three times selected for the two-aircraft case, while the green dashed lines mark the three times selected for the three-aircraft case. MAC risk values outside these selected times result from other qualifying aircraft pairs present during the ten-minute interval and are retained to provide temporal context.

Centroid 3 generally exhibits higher normalized MAC risk values than centroid 4 over the displayed interval and within the selected cases. Under the proposed metric, this indicates that qualifying aircraft pairs

generally produce shorter and more closely matched convergence times toward the lower-altitude location represented by centroid 3 than toward the 2,400 m location represented by centroid 4. This reflects a stronger local aircraft-convergence condition at centroid 3 for the aircraft geometries represented in this interval.

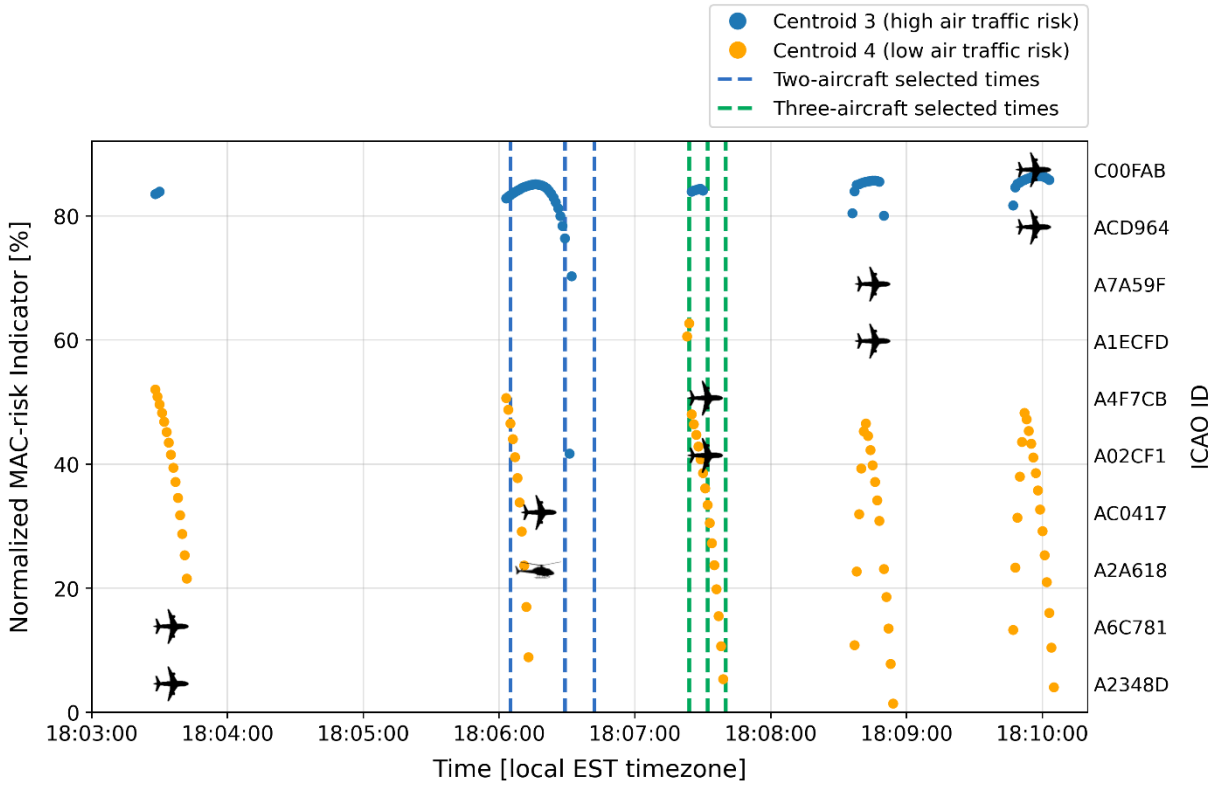


Figure 66: MAC risk metric over time for centroids 3 and 4

6.4.2.1 Two-aircraft scenario

The two-aircraft scenario examines aircraft A2A618 and AC0417, which are simultaneously present during the selected interval from approximately 18:06:05 to 18:06:42. Aircraft A2A618 is an Airbus AS350 helicopter operating at a relatively low altitude (~1,000 m) over the LGA area, while aircraft AC0417 is a United Airlines ERJ 170-200 LR twin-engine regional jet approaching runway 13. Figures 67-71 show the resulting location-based MAC risk field at 18:06:05, 18:06:29, and 18:06:42, corresponding to the blue dashed lines in Figure 66.

Figures 67-69 show the three-dimensional normalized MAC risk indicator field as the trajectories of A2A618 and AC0417 evolve at the three selected times. At 18:06:05, non-zero indicator values occupy a broad volume of the model, with the largest values concentrated in the region toward which the two aircraft are converging. The non-zero volume subsequently changes in position and contracts as the relative aircraft geometry evolves through 18:06:29 and 18:06:42.

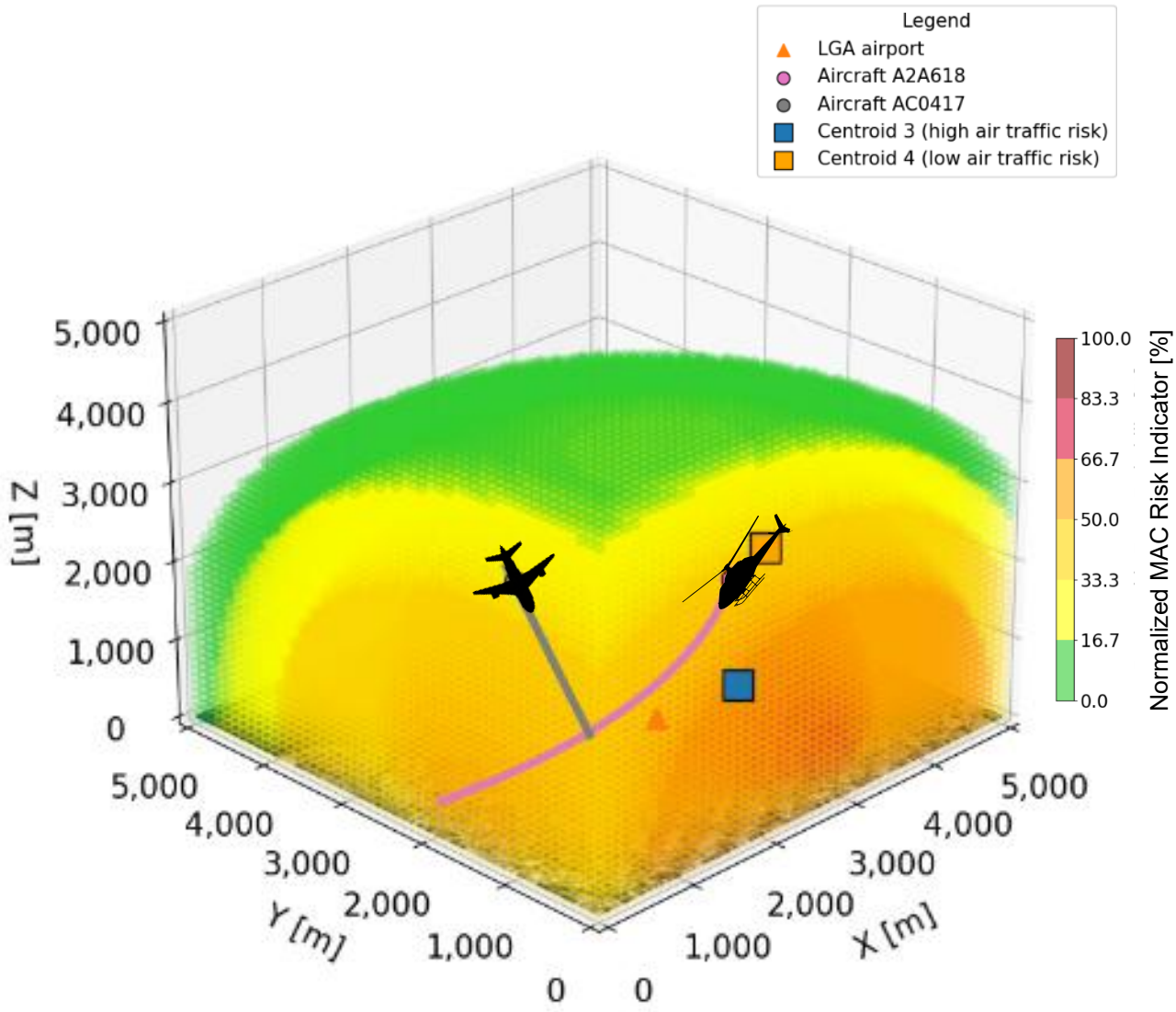


Figure 67: 3D normalized MAC risk indicator field for the two-aircraft scenario at 2022-11-22 18:06:05

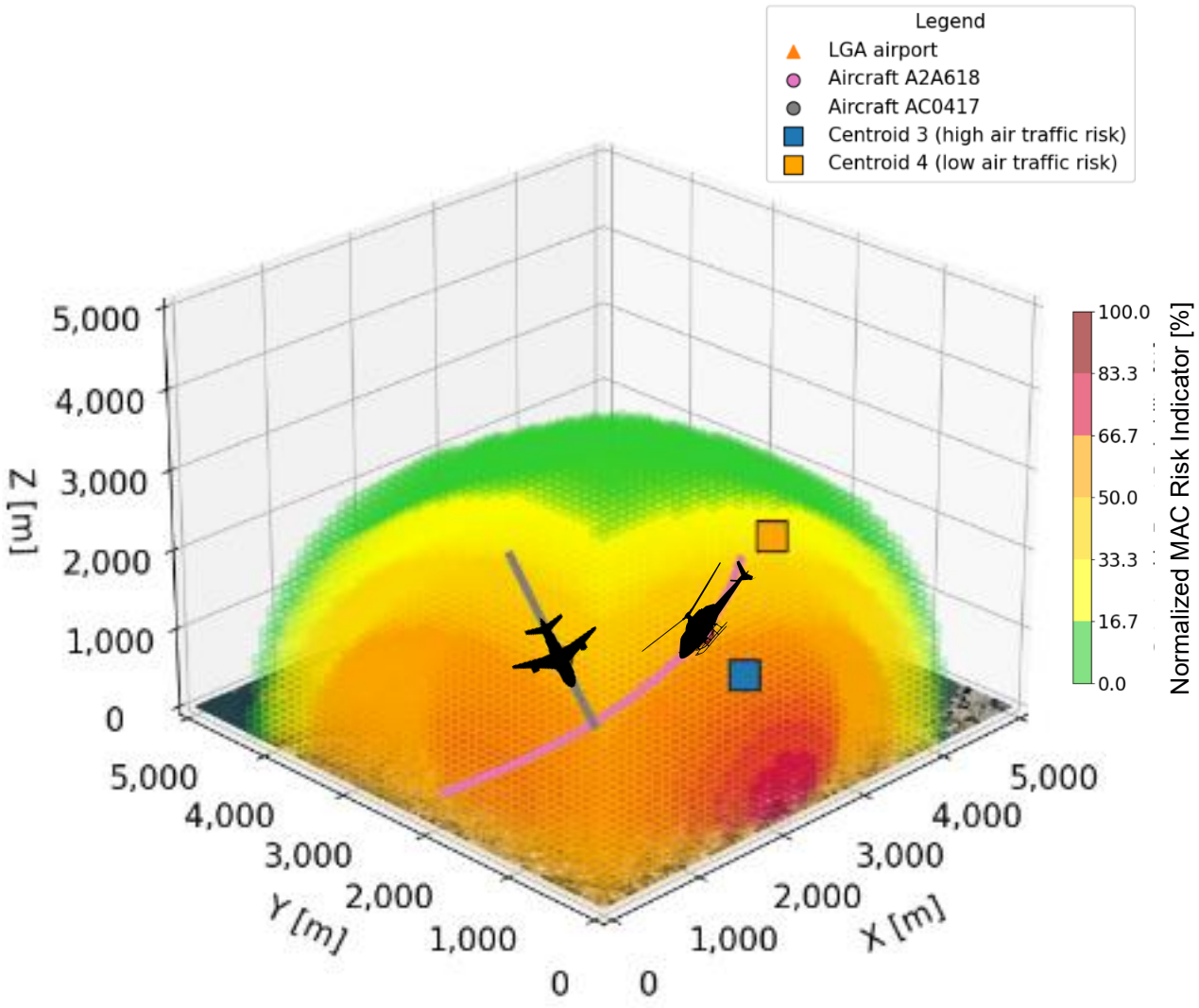


Figure 68: 3D normalized MAC risk indicator field for the two-aircraft scenario at 2022-11-22 18:06:29

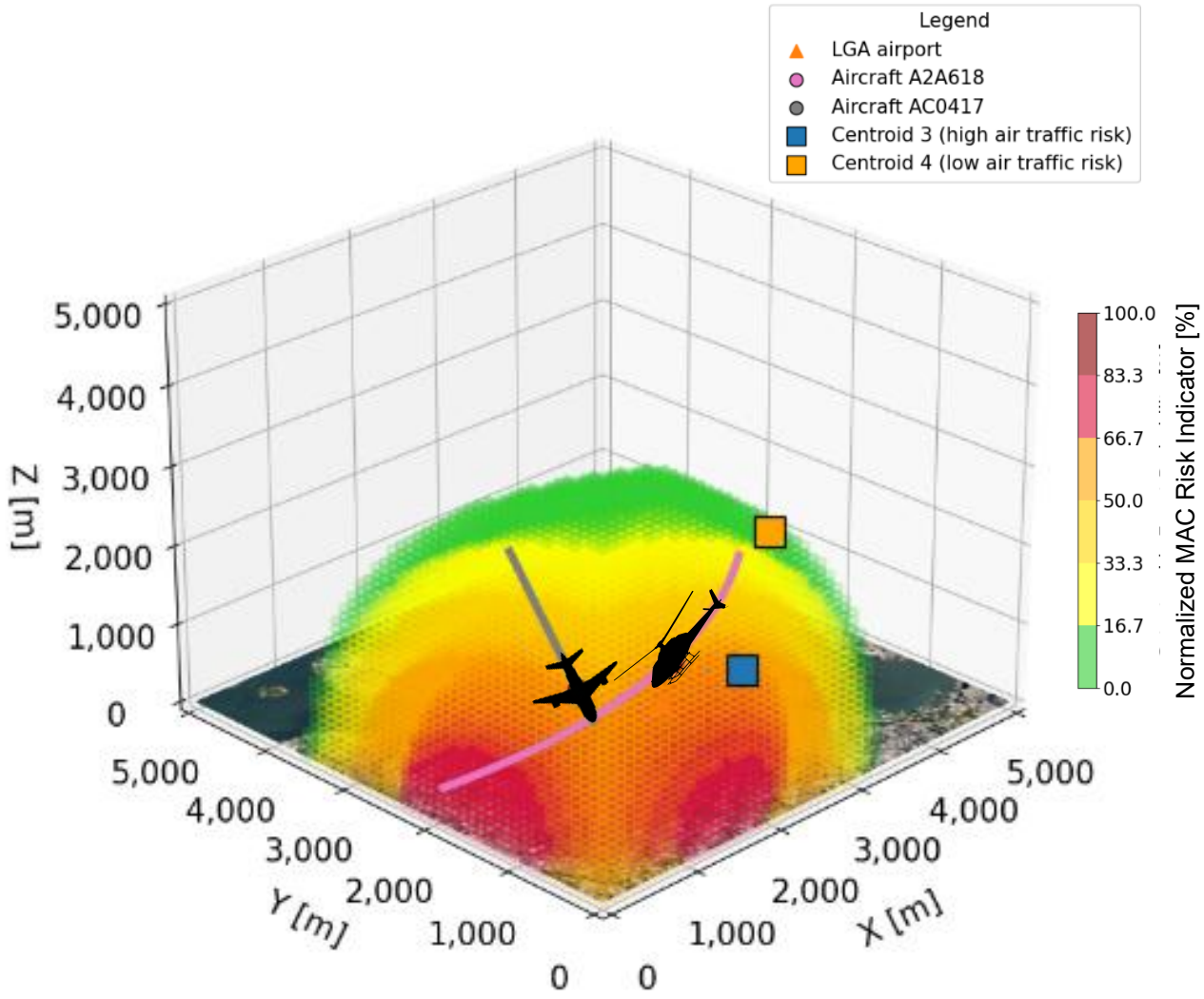
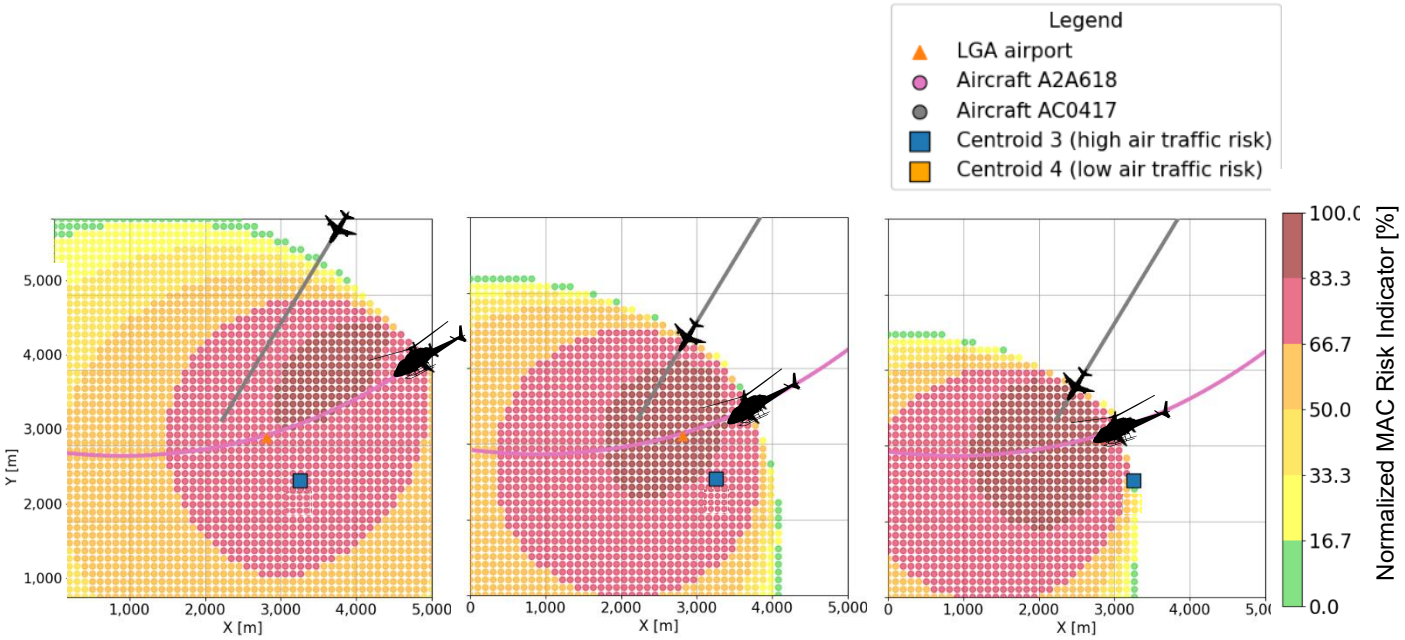


Figure 69: 3D normalized MAC risk indicator field for the two-aircraft scenario at 2022-11-22 18:06:42

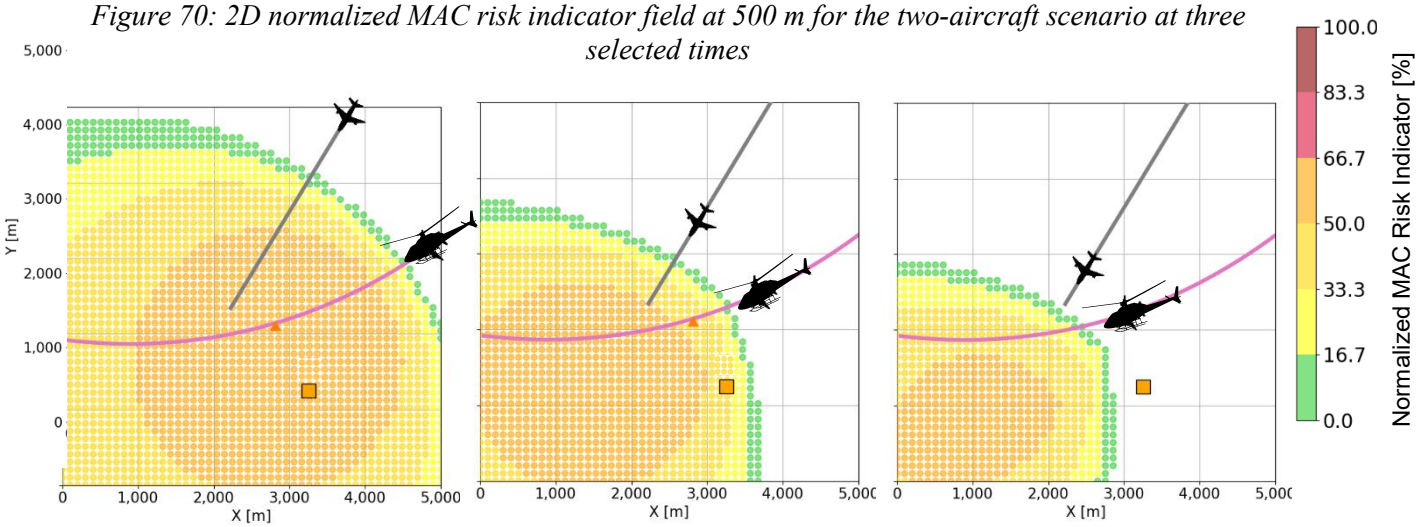
Figure 70 presents 2D horizontal slices of the normalized MAC risk indicator field at 500 m, corresponding to the altitude of centroid 3. At 18:06:05, much of the central and eastern portions of the displayed field falls within the 67–100% indicator bands, including a localized region in the 83–100% band. The position and extent of the higher indicator values change at 18:06:29 and 18:06:42 as the aircraft trajectories and their convergence geometry evolve. These spatial changes provide context for the MAC risk indicator values observed at centroid 3 in Figure 66.

Figure 71 presents the corresponding 2D horizontal slices at 2,400 m, corresponding to the altitude of centroid 4. The normalized MAC risk indicator field is generally lower at this altitude: at 18:06:05, most non-zero locations fall within the 17–67% bands, with no comparably large region in the 83–100% band. The non-zero field contracts further at 18:06:29 and 18:06:42. Comparison with Figure 70 shows that the same aircraft pair produces different spatial distributions of the indicator at 500 m and 2,400 m because the three-dimensional distances and convergence geometry relative to the evaluated DGGS locations vary with altitude.



a) 2022-11-22 18:06:05 b) 2022-11-22 18:06:29 c) 2022-11-22 18:06:42

Figure 70: 2D normalized MAC risk indicator field at 500 m for the two-aircraft scenario at three selected times



a) 2022-11-22 18:06:05 b) 2022-11-22 18:06:29 c) 2022-11-22 18:06:42

Figure 71: 2D normalized MAC risk indicator field at 2,400 m for the two-aircraft scenario at three selected times

6.4.2.2 Three-aircraft scenario

The three-aircraft scenario extends the same MAC risk calculation to three selected aircraft, allowing more than one qualifying aircraft pair to contribute to the location-based field at a given time. This provides a controlled increase in pairwise convergence complexity relative to the two-aircraft scenario in Section 6.4.2.1.

The selected aircraft are A2A618 (Airbus AS350 helicopter), A02CF1 (private ERJ 170-200 LR twin-engine regional jet), and A4F7CB (Republic Airways ERJ 170-200 LR twin-engine regional jet). Their trajectories are shown consistently in pink, dark yellow, and cyan, respectively, in Figures 72-76. The three selected times—18:07:24, 18:07:32, and 18:07:40—correspond to the green dashed lines in Figure 66.

Figures 72-74 show the three-dimensional normalized MAC risk indicator field for the three-aircraft scenario. Because more than one aircraft pair can satisfy the convergence criteria, distinct non-zero regions can occur at different DGGS locations at the same time. At 18:07:24, separate regions are visible on different sides of the model. Their position and extent change at 18:07:32 and 18:07:40 as A02CF1, A2A618, and A4F7CB move relative to one another. The resulting field is therefore spatially more complex than in the selected two-aircraft case because different qualifying pairs can determine the local MAC risk indicator at different locations.

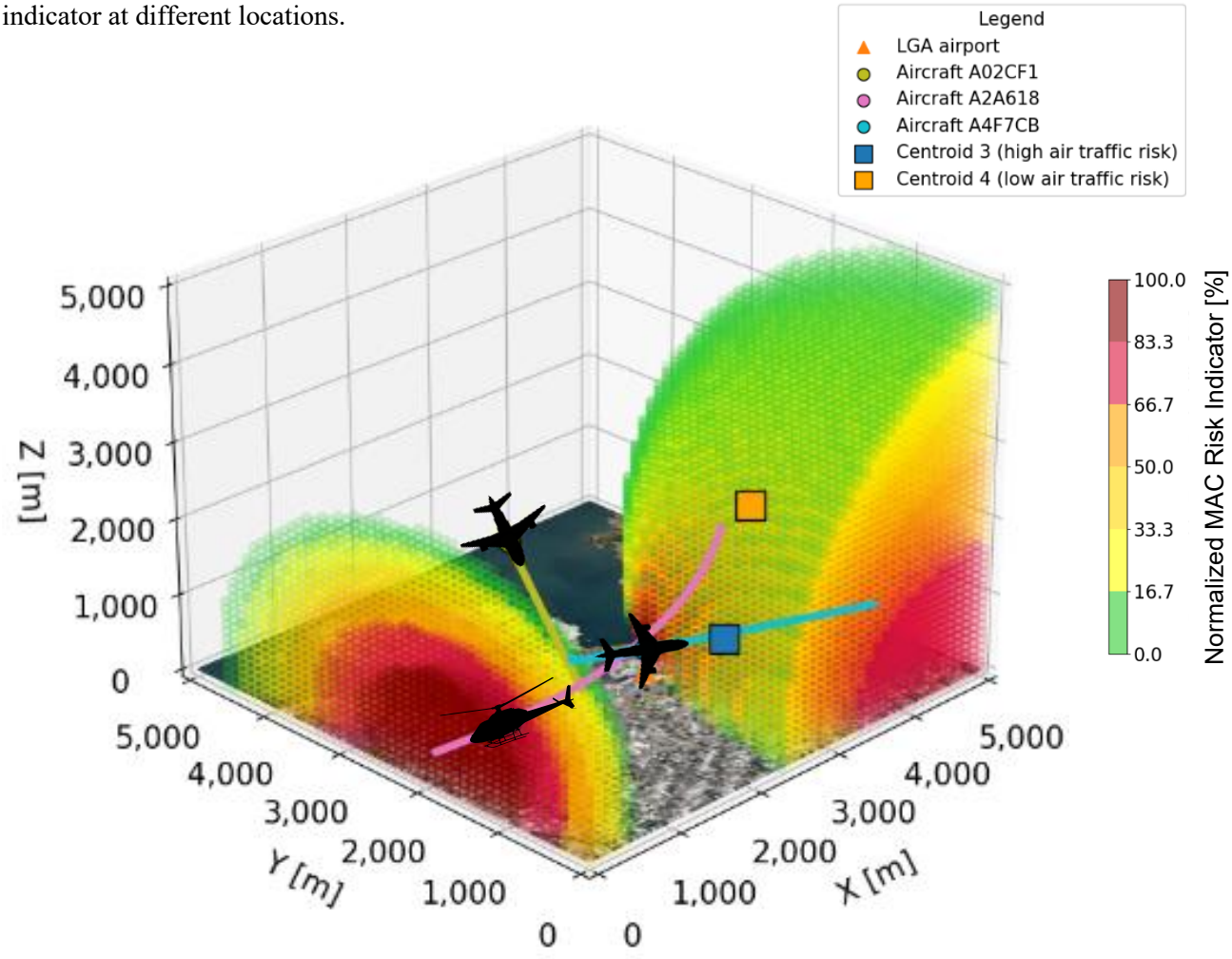


Figure 72: 3D normalized MAC risk indicator field for the three-aircraft scenario at 2022-11-22 18:07:24

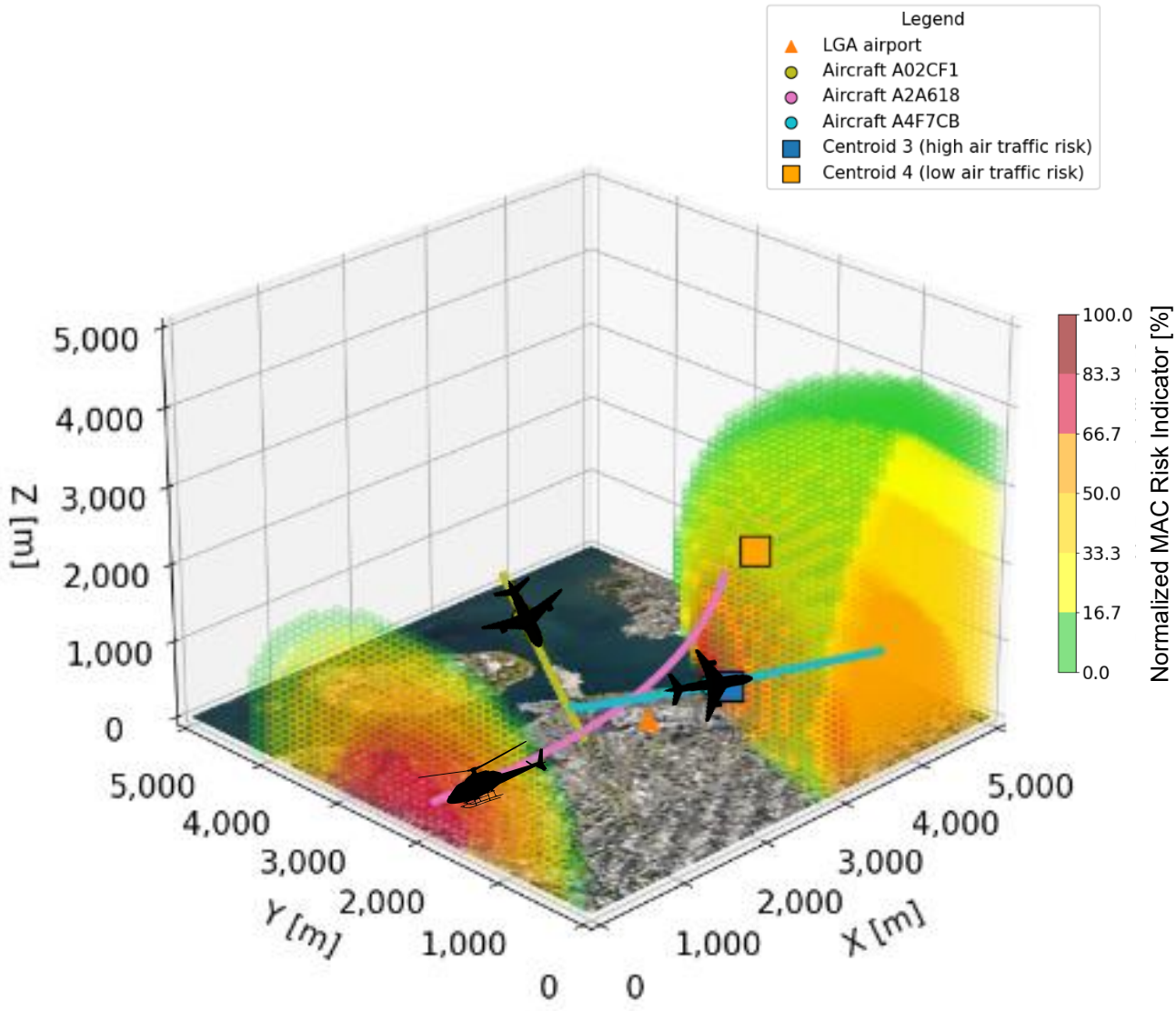


Figure 73: 3D normalized MAC risk indicator field for the three-aircraft scenario at 2022-11-22 18:07:32

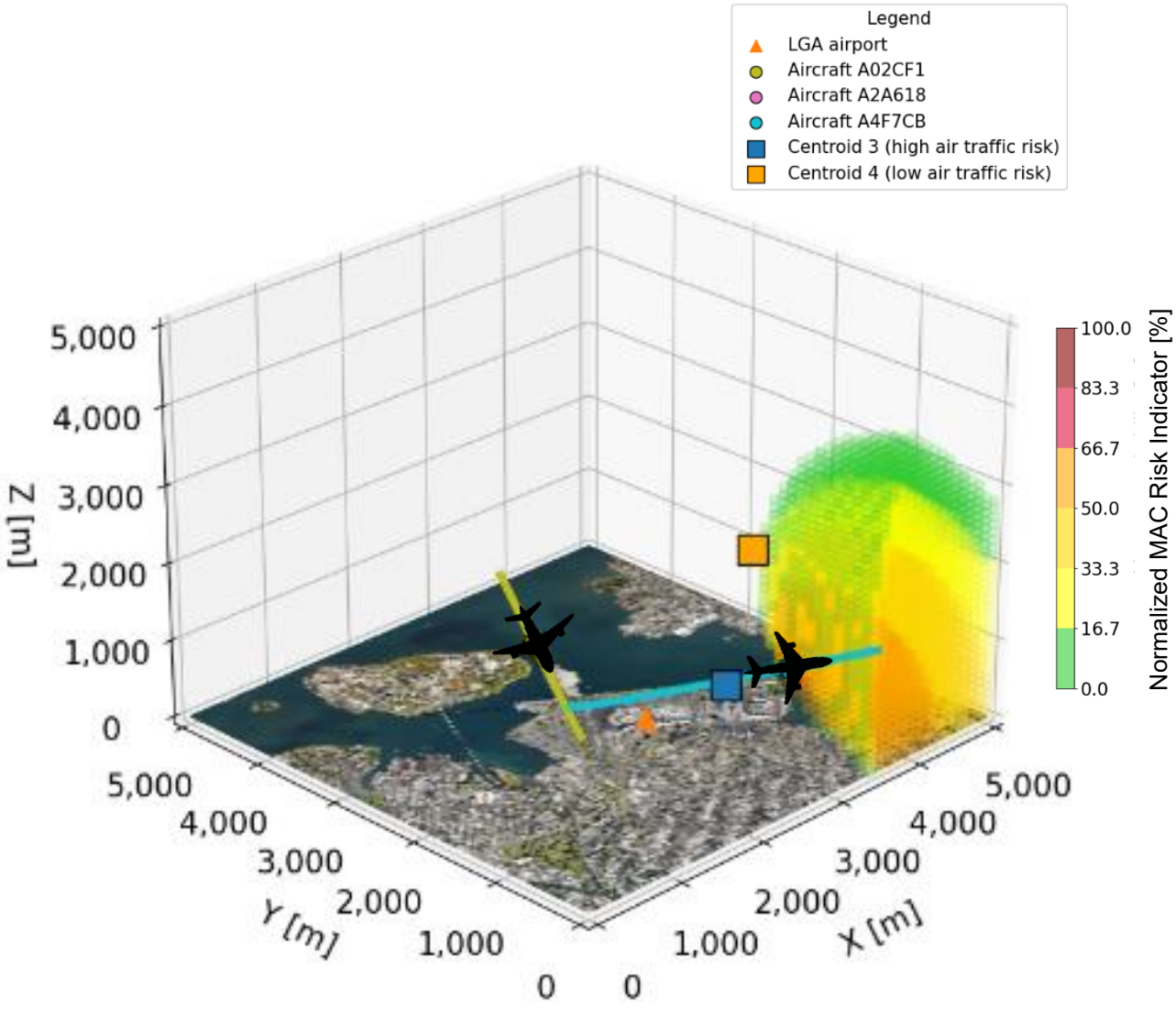
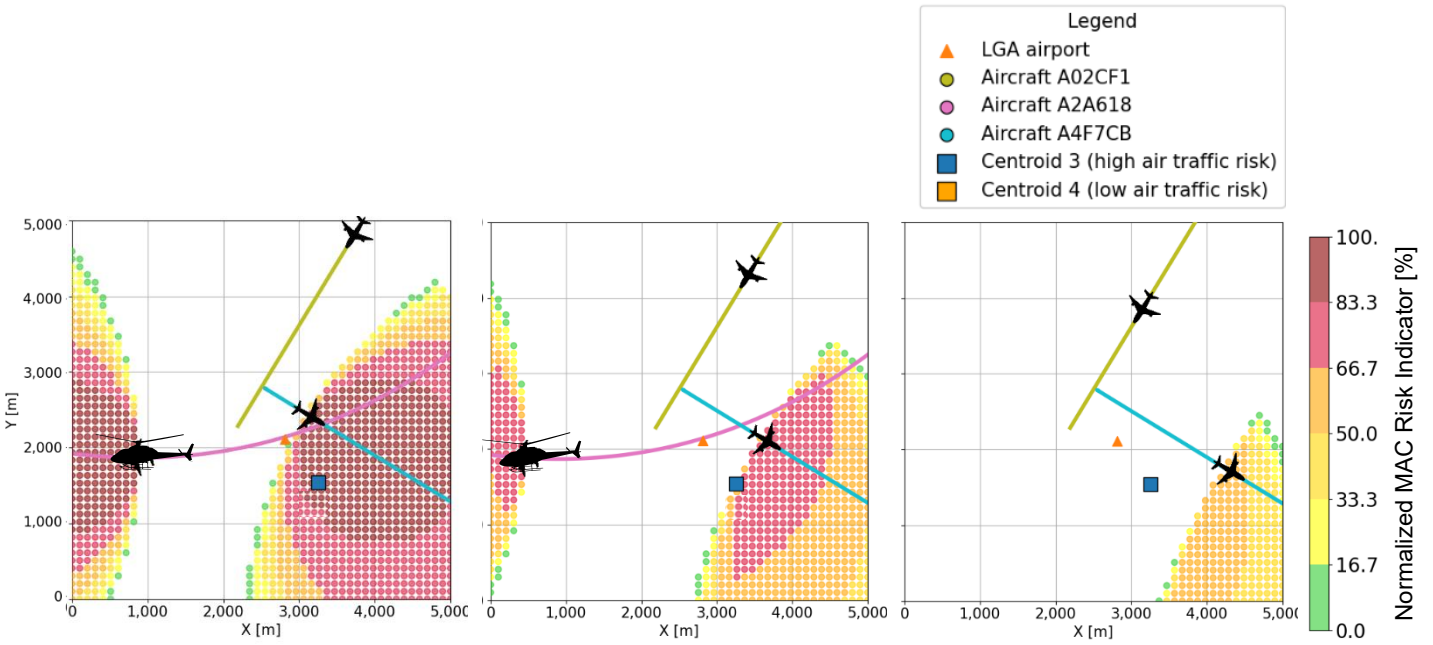


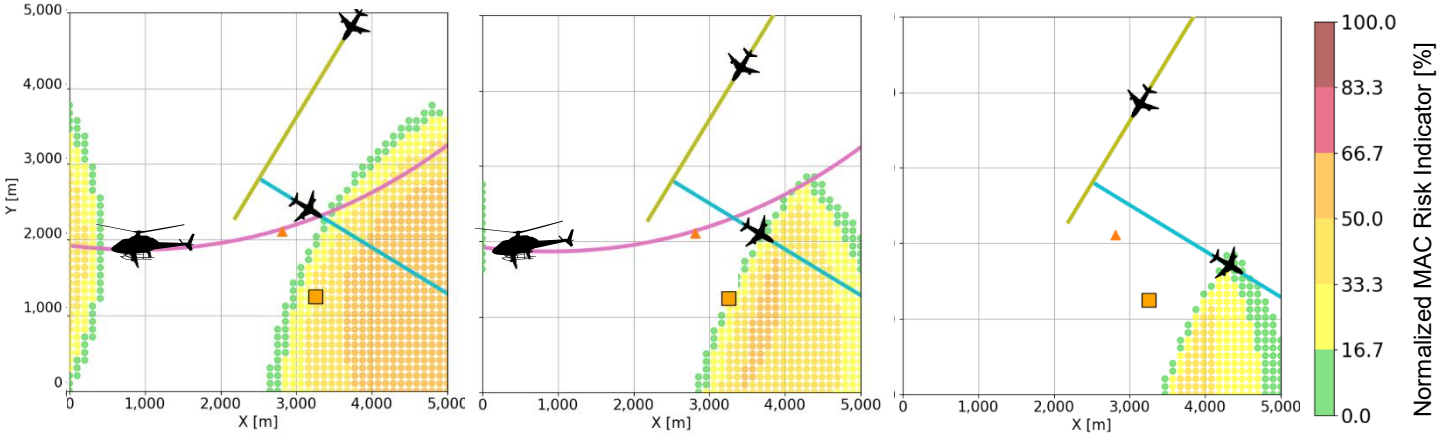
Figure 74: 3D normalized MAC risk indicator field for the three-aircraft scenario at 2022-11-22 18:07:40

Figures 75-76present 2D horizontal slices of the same field at 500 m and 2,400 m, respectively. At 500 m in Figure 75, the 18:07:24 field contains two separated regions with indicator values extending into the 67–100% bands. By 18:07:40, the non-zero field is concentrated primarily near the eastern side of the model and is dominated by lower indicator bands. At 2,400 m in Figure 76, the non-zero regions are smaller and generally occupy lower indicator bands than in the corresponding 500 m slices. These results show that multiple aircraft pairs can contribute to spatially distinct convergence regions while the resulting field remains dependent on the three-dimensional DGGS location at which the indicator is evaluated.



a) 2022-11-22 18:07:24 b) 2022-11-22 18:07:32 c) 2022-11-22 18:07:40

Figure 75: 2D normalized MAC risk indicator field at 500 m for the three-aircraft scenario at three selected times

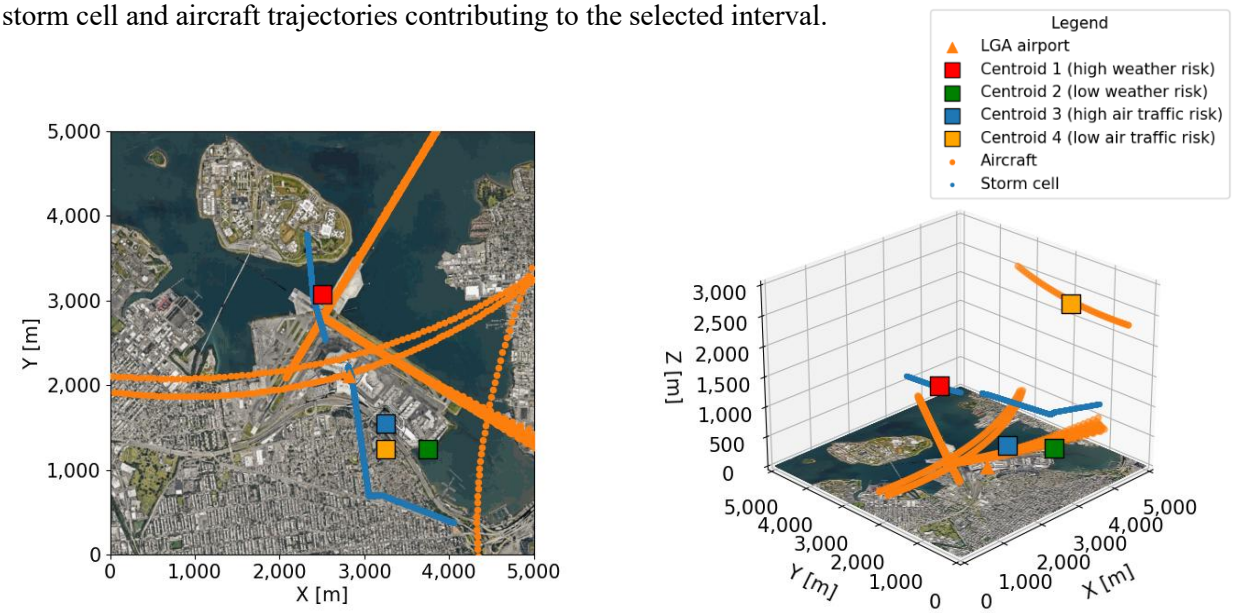


a) 2022-11-22 18:07:24 b) 2022-11-22 18:07:32 c) 2022-11-22 18:07:40

Figure 76: 2D normalized MAC risk indicator field at 2,400 m for the three-aircraft scenario at three selected times

6.4.3 Combined weather–MAC risk results

This section presents the combined weather–MAC risk results. The same four reference centroids examined in Sections 6.4.1 and 6.4.2 are retained so that the individual weather- and MAC risk indicators and their combined result can be compared at common DGGS locations. Figure 77 provides spatial context for the storm cell and aircraft trajectories contributing to the selected interval.



a) 2D satellite view

b) 3D satellite view

Figure 77: Centroids 1, 2, 3, and 4 with storm and air traffic data

Figure 78 shows the normalized combined weather–MAC risk indicators for centroids 1–4 during the selected case-study interval. The red, green, blue, and orange curves represent the temporal evolution of the combined risk indicator at centroids 1, 2, 3, and 4, respectively. The secondary vertical axis identifies the storm cells and aircraft associated with the analysis. Storm icons indicate the weather cell context over the selected interval, while aircraft icons identify those involved in the MAC risk calculations at the selected centroids, with their horizontal positions indicating the corresponding periods of involvement. Three times are selected for detailed spatial examination: 18:06:00, when the combined indicator reflects weather risk alone; 18:06:06, when both weather and MAC risks are present; and 18:06:54, when the selected field again reflects only the weather risk component. These times are marked by the blue dashed vertical lines and correspond to the spatial fields presented in Figures 79–85.

At 18:06:00, storm cells 0 and 1 influence the weather risk indicator, while no qualifying aircraft pair generates a MAC risk value, so the combined field corresponds to the weather risk field. At 18:06:06, the weather risk component remains present, while aircraft A2A618 and AC0417 generate a MAC risk value, producing a combined field that incorporates both sources of risk. At 18:06:54, no qualifying aircraft pair is identified for the selected field, and the result again corresponds to the weather risk component. The temporal profiles in Figure 78 also illustrate that the combined risk indicator varies between the four centroids according to their respective spatial relationships with the relevant entities. The sources influencing the risk assessment can therefore change over time while the analytical quantities remain referenced to the same DGGS locations and time structure.

This progression demonstrates an important property of the common location-based representation: the analytical field does not require a different spatial definition when the underlying risk sources change. Weather-only, MAC-only, and simultaneous risk conditions can all be represented using the same DGGS location and time reference, allowing their respective effects to be examined within a consistent spatiotemporal framework.

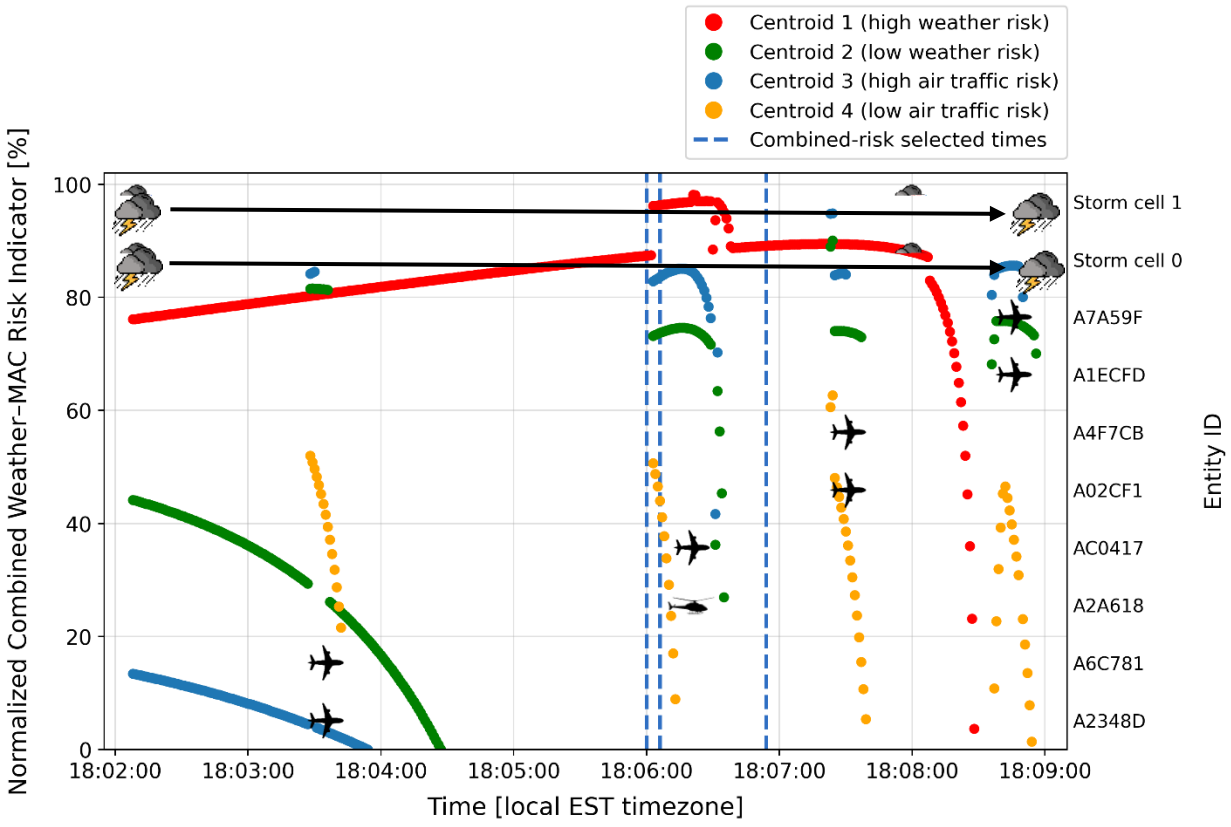


Figure 78: Combined risk metric over time for centroids 1, 2, 3, and 4

Figures 79-81 present the complete three-dimensional combined weather–MAC risk indicator field at the three selected times. Figures 82-85 present horizontal slices of the same field at 1,200 m, 300 m, 500 m, and 2,400 m, respectively. These altitudes correspond to the reference altitudes of centroids 1–4, but each figure represents the combined indicator across all DGGS locations in the displayed altitude plane rather than a calculation performed only at the named centroid.

At 18:06:00, the spatial distribution reflects the weather risk component. At 18:06:06, the additional MAC risk contribution changes the magnitude and spatial distribution of the combined field; this is particularly visible in the 1,200 m slice of Figure 82, where a larger portion of the central region occupies the higher normalized indicator bands. By 18:06:54, the MAC risk contribution is absent and the combined field again follows the weather-derived spatial pattern associated with storm cells 0 and 1.

Together, these results demonstrate that independently calculated indicators derived from aircraft and weather data can be co-indexed and combined at common three-dimensional DGGS locations and times without reconstructing the underlying airspace representation.

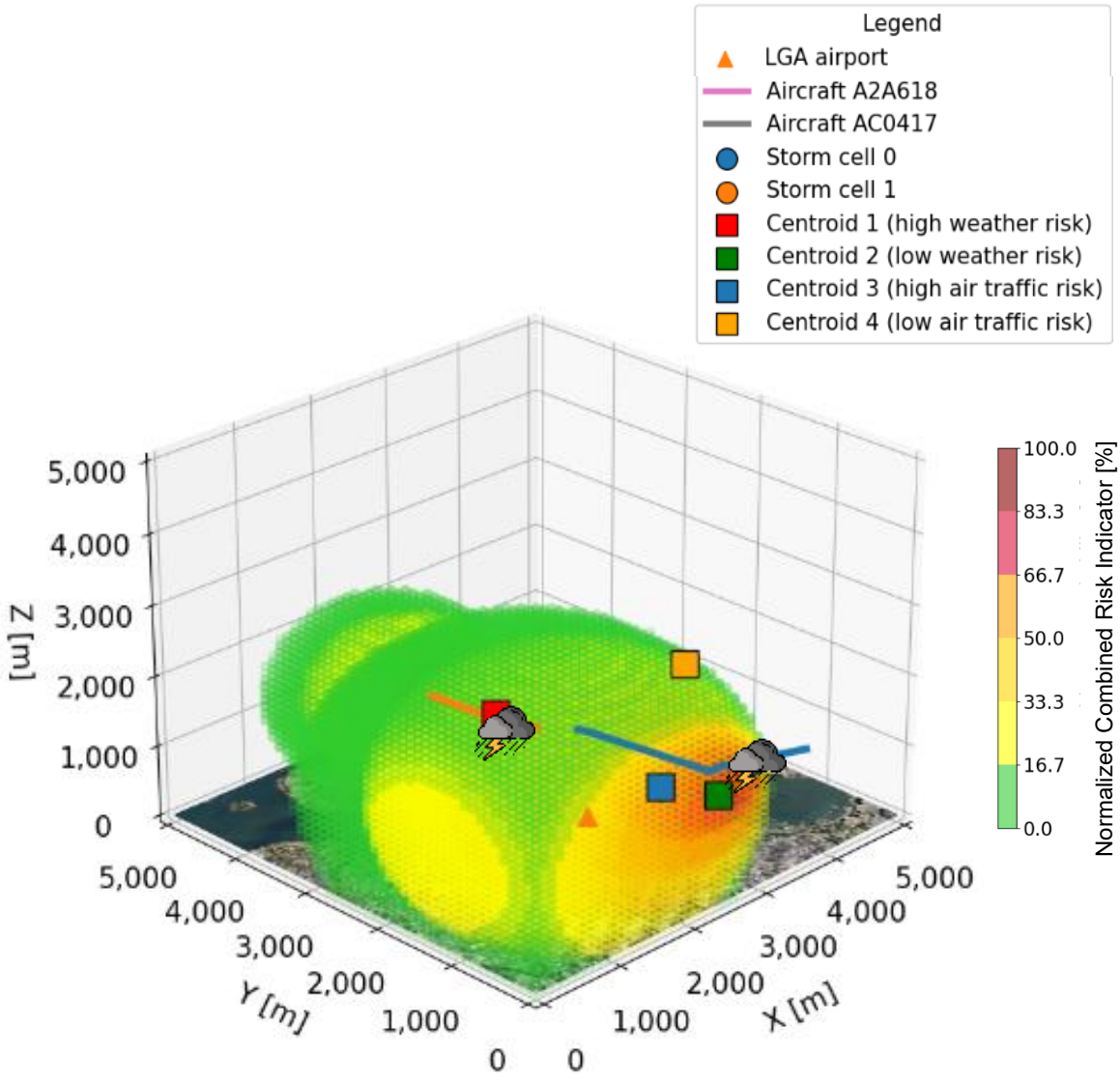


Figure 79: 3D normalized combined weather–MAC risk indicator field at 2022-11-22 18:06:00

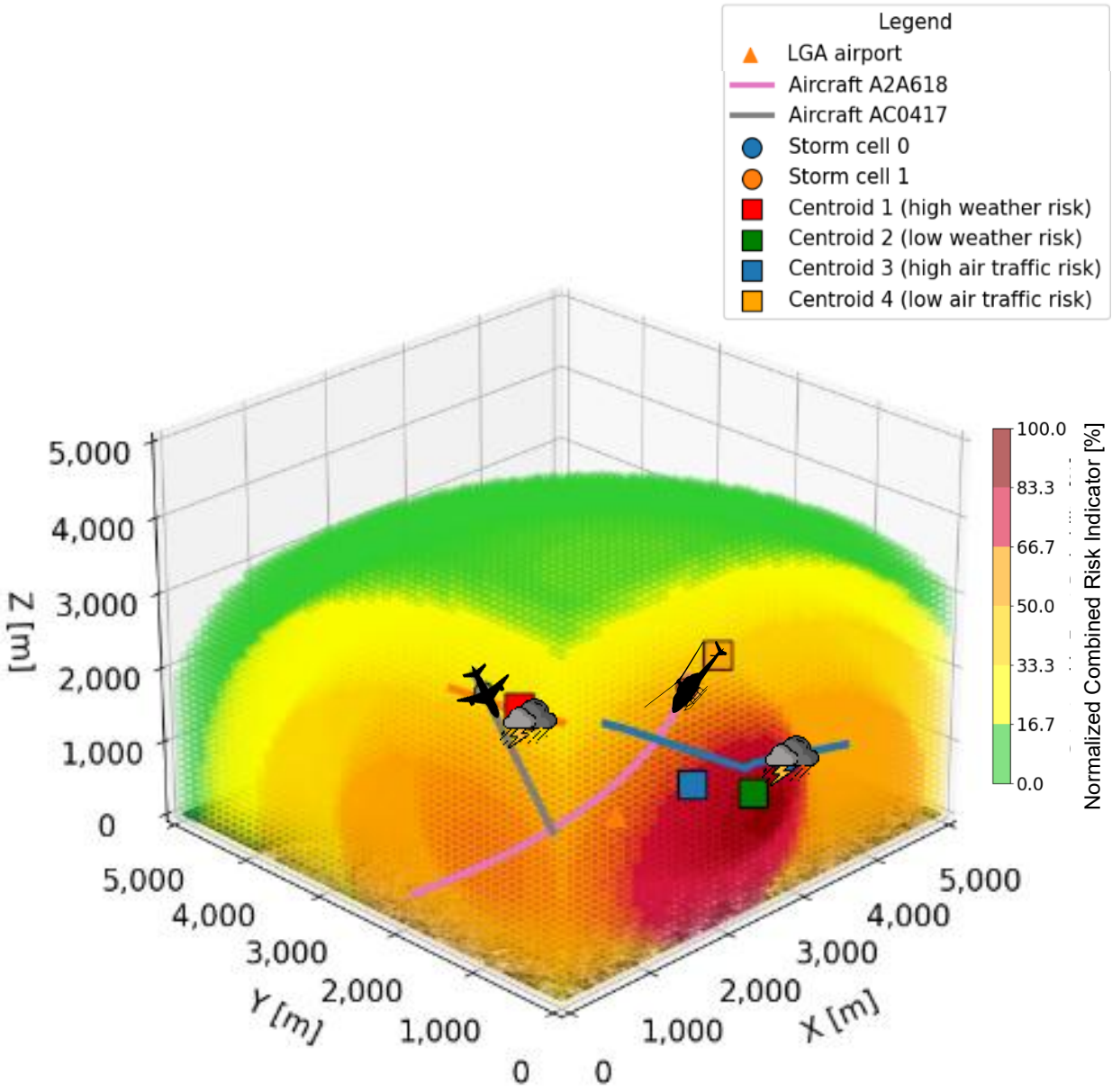


Figure 80: 3D normalized combined weather–MAC risk indicator field at 2022-11-22 18:06:06

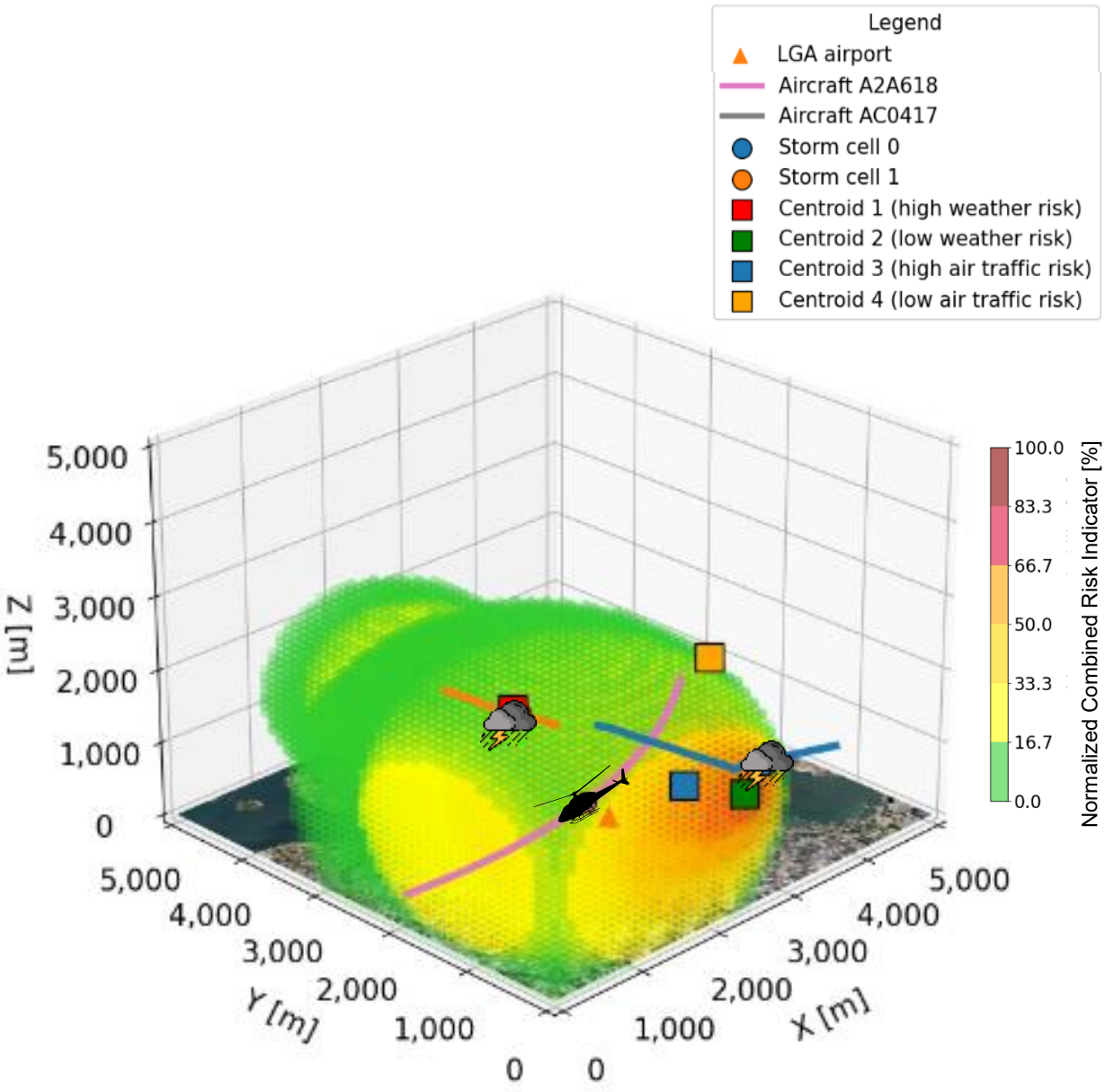


Figure 81: 3D normalized combined weather–MAC risk indicator field at 2022-11-22 18:06:54

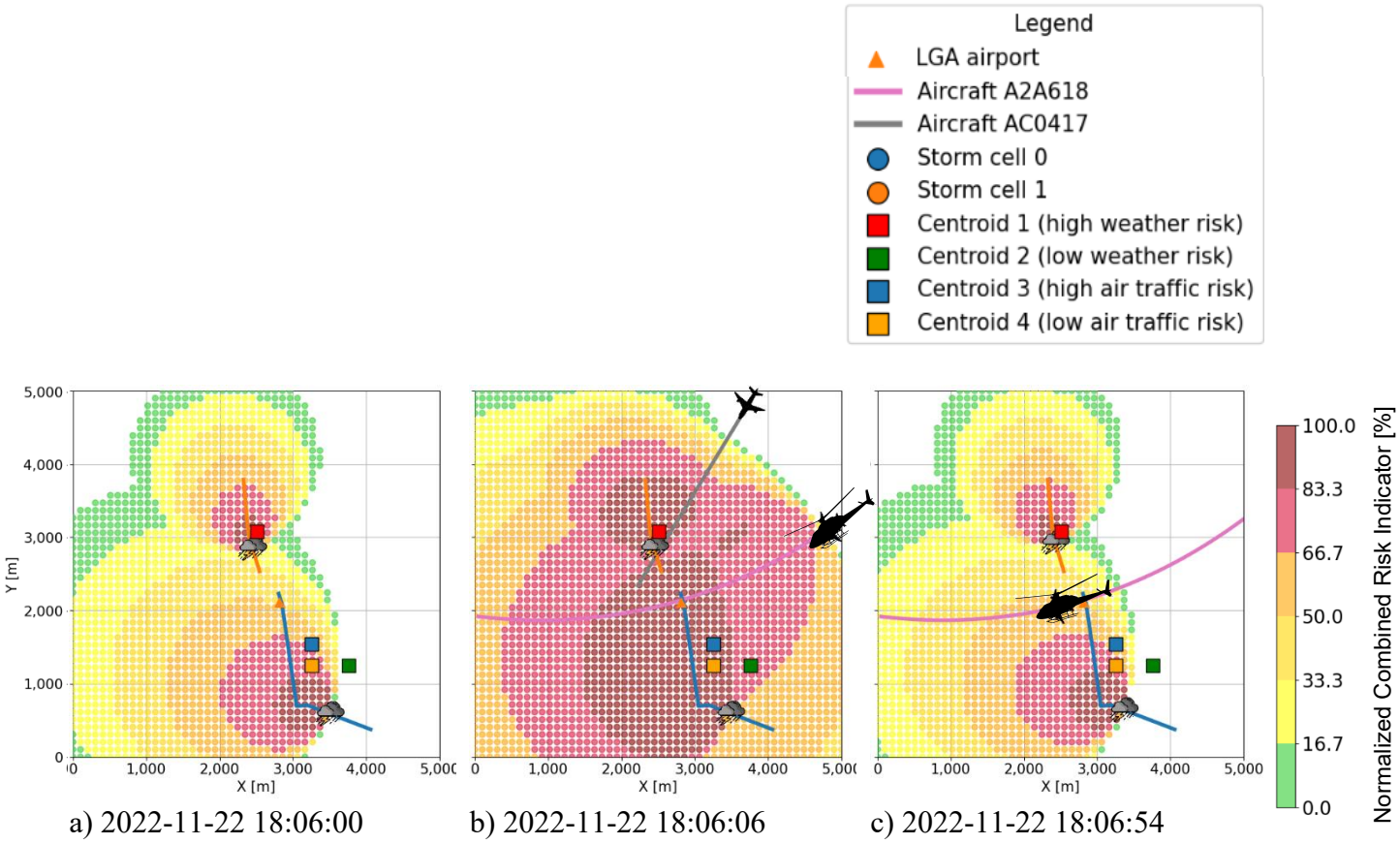


Figure 82: 2D combined weather–MAC risk indicator field at 1,200 m at three selected times

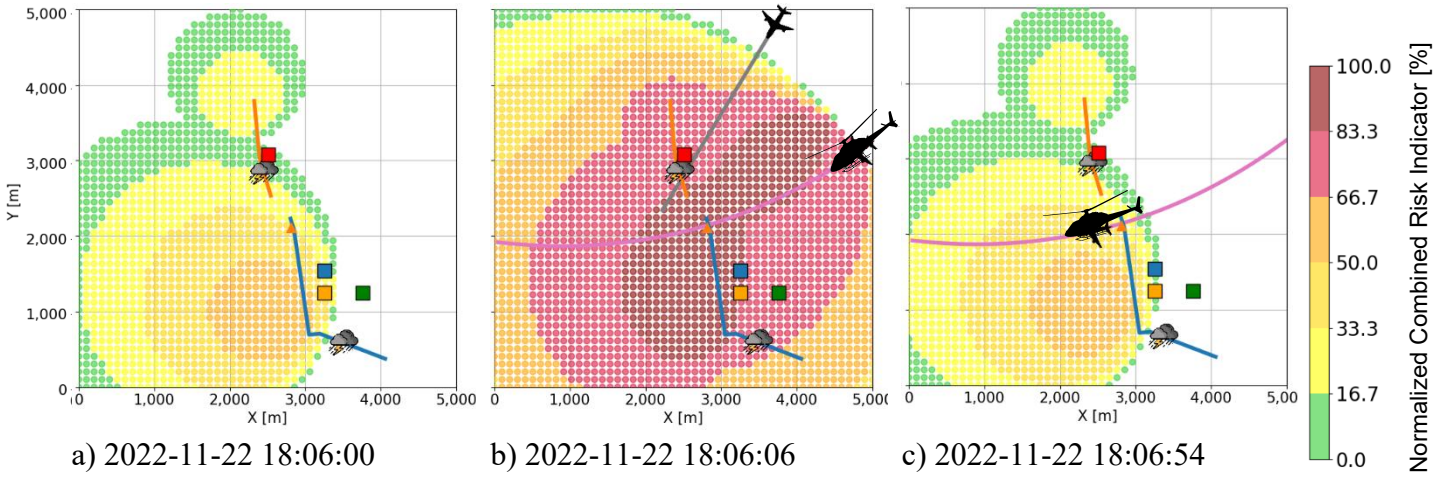
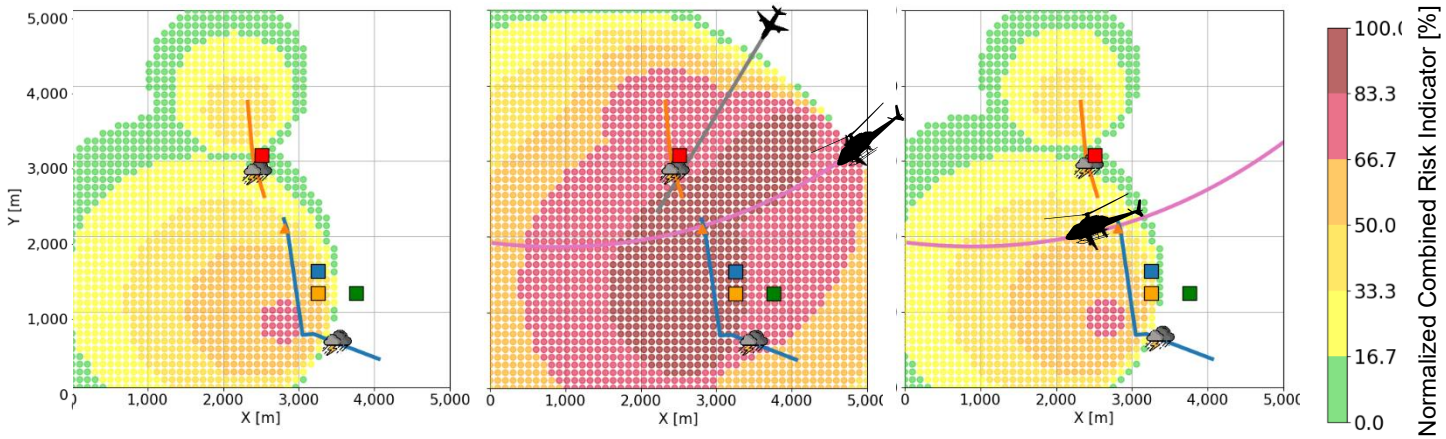
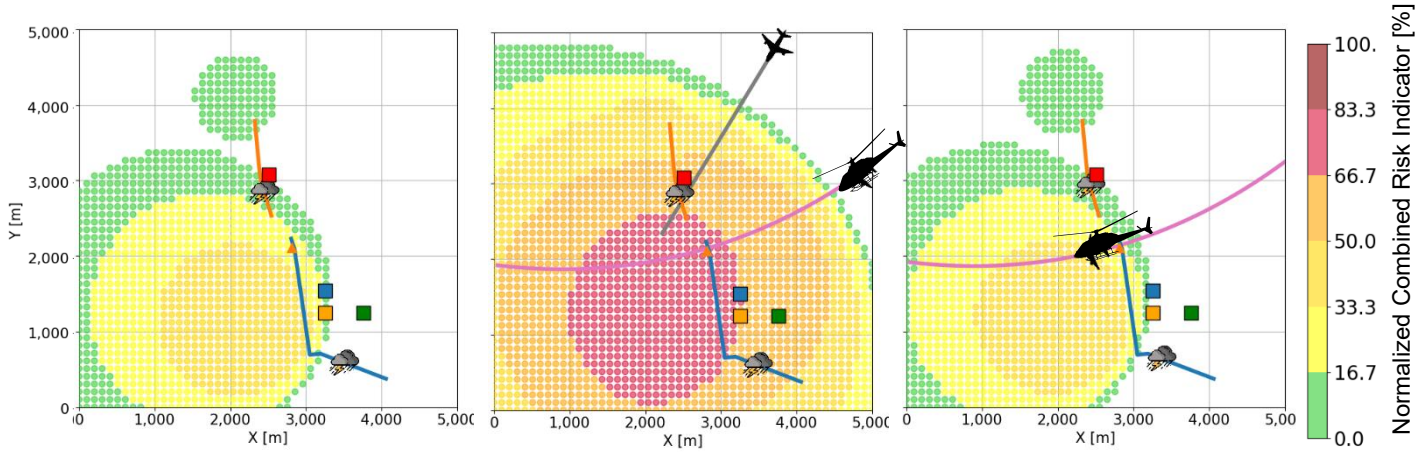


Figure 83: 2D combined weather–MAC risk indicator field at 300 m at three selected times



a) 2022-11-22 18:06:00 b) 2022-11-22 18:06:06 c) 2022-11-22 18:06:54

Figure 84: 2D combined weather–MAC risk indicator field at 500 m at three selected times



a) 2022-11-22 18:06:00 b) 2022-11-22 18:06:06 c) 2022-11-22 18:06:54

Figure 85: 2D combined weather–MAC risk indicator field at 2,400 m at three selected times

6.5 Experiment #2 discussion

Experiment #2 evaluated the **location-based metric computation branch** of the proposed airspace-modelling methodology. Whereas Experiment #1 established the reusable DGGS-based representation of heterogeneous historical airspace data, Experiment #2 examined whether information contained within that representation could subsequently be transformed into new location-based analytical products. Risk analysis served as the case study, with hazardous-weather and aircraft-proximity observations providing two physically distinct, time-varying data sources from which separate and combined risk metrics were derived.

The main result is that the existing spatial and temporal reference of the airspace model can be retained during this transformation. Weather- and MAC risk metrics were computed independently at DGGS locations and time steps, produced interpretable spatial and temporal variation consistent with their respective source observations, and were then combined using the same reference structure. The same spatial model and reference structure were retained for aircraft and weather risk throughout the analysis, allowing both components and their combination to be evaluated using the existing airspace representation.

This demonstrates the transition from airspace-model construction to analytical reuse. Heterogeneous historical observations are first organized within the common 4D airspace model and can subsequently be transformed into new metrics associated with fixed three-dimensional locations and times. Risk is used here as the selected case study rather than as the sole intended application of the methodology. Other applications could define different metrics over the same model contents—for example, traffic density, congestion, environmental exposure, occupancy, or other operational measures—provided that the required source variables are available. Experiment #2 demonstrates this underlying analytical mechanism through a controlled multi-source case study, while extension to other analytical products remains available for future applications of the methodology.

6.5.1 Interpretation of the location-based risk metric results

6.5.1.1 Weather risk results

The weather-only results demonstrate that moving storm cell observations can be transformed into risk metrics evaluated at fixed DGGS locations. Risk is calculated independently at each three-dimensional centroid. As a storm cell moves, its three-dimensional distance and convergence rate relative to each centroid evolve differently. Centroids 1 and 2 occupy different spatial locations and different altitudes—1,200 m and 300 m, respectively—so the same moving storm cell dataset produces different convergence times and therefore different risk indicator histories at the two locations.

The important analytical distinction is that the output goes beyond mapping storm cell locations by assigning an analytical value to fixed airspace locations based on the estimated convergence of the storm cells toward them. The weather observations are therefore transformed into a new location-based analytical product while retaining their relationship to three-dimensional position and time for the DGGS reference.

6.5.1.2 Location-based MAC results

The aircraft-only results demonstrate the same analytical transformation but using a fundamentally different source dataset and calculation. ADS-B aircraft trajectories are converted into a fixed-location field that identifies where qualifying aircraft pairs converge toward the same three-dimensional DGGS location and have similar estimated arrival times.

The two-aircraft scenario provides the clearest physical interpretation. A higher risk indicator at centroid 3 than at centroid 4 indicates that the two aircraft are projected to reach centroid 3 sooner and with a smaller difference between their estimated arrival times. Under the proposed metric, centroid 3 therefore represents the more immediate common convergence location for this aircraft pair at that time.

The three-aircraft scenario shows that different aircraft pairs can control the metric at different locations simultaneously. Although this produces a more complex spatial field, each risk value remains associated with an explicit DGGS location and time, and can therefore be examined and interpreted using the same location-based framework.

6.5.1.3 Combined risk results

The combined results provide the strongest evidence of the common-reference capability tested in Experiment #2. The weather- and MAC risk metrics originate from different source datasets, use different physical calculations, and evolve over different time scales. Nevertheless, because both are referenced to common DGGS locations and time steps, they can be examined independently and subsequently combined without introducing a separate spatial representation.

The sequence at 18:06:00, 18:06:06, and 18:06:54 illustrates this directly. The same fixed locations transition from weather-only contributions to simultaneous weather and MAC contributions, and back to weather-only contributions as the underlying entities change. The analytical significance is therefore not only that a combined risk value can be calculated, but also that the identity and contribution of heterogeneous risk sources can change while the underlying spatial and temporal reference remains unchanged.

6.5.2 Validation of the Experiment #2 location-based risk model requirements

The six requirements introduced in Section 6.2.1 are used to determine whether the characteristics needed for the location-based risk case study persist after the contents of the reusable airspace model are transformed into derived risk metrics. This is an important distinction in the validation logic of Experiment #2. Experiment #1 demonstrated that the underlying model can represent heterogeneous aircraft and weather observations in three dimensions and over time, at different model scales and DGGS resolutions. Experiment #2 therefore evaluates whether these modelling characteristics remain available when the model contents are transformed into a derived analytical product. Complete validation of the intended risk application requires the resulting fields to retain explicit altitude, temporal variation, and common spatial and temporal referencing of the aircraft- and weather-derived quantities, in addition to producing the corresponding numerical risk indicators.

The strength of the experimental evidence varies across the six requirements. Several are demonstrated directly by the results of Experiment #2, whereas applicability to different geographic regions and large-scale computational scalability are supported primarily by the general formulation and by results from Experiment #1. Table 24 summarizes this evidence.

Table 24: Validation of Experiment #2 results against the location-based risk model requirements

Requirement model requirement	Evidence from the results
Air-vehicle agnostic	The same MAC risk formulation is applied to a helicopter and fixed-wing aircraft using observed trajectory position, altitude, time, and motion rather than vehicle-specific aerodynamic models.
Applicable to different airspace regions	The metric is defined relative to DGGS locations rather than a fixed route or sector geometry. Experiment #1 demonstrates that the underlying modelling workflow can be applied across different domain sizes and resolution strategies. Independent regional replication of the risk formulation of Experiment #2 is reserved for future validation.
Three-dimensional	Weather- and MAC risk results differ between selected altitude levels, demonstrating that altitude is retained explicitly in the derived risk fields.
Time-varying	Risk values change as the source entities move. The weather example evolves over approximately eight minutes, whereas the selected aircraft MAC scenarios change substantially in their risk indicators within a few seconds.
Scalable in space, time, and entity count	The same calculations apply across many DGGS locations and time steps, and the MAC case is extended from one aircraft pair to several candidate pairs. Large-scale computational scaling of the underlying model is evaluated separately in Experiment #1.
Multi-hazard integration	Weather- and MAC risk metrics are independently derived and subsequently combined at the same DGGS locations and times.

Table 24 shows that the results of Experiment #2 demonstrate that four of the requirements are met: air-vehicle agnosticism for the selected aircraft, three-dimensional representation, time-varying analysis, and multi-hazard integration. Applicability across geographic regions is supported by the general location-based formulation and by Experiment #1, which demonstrates the underlying modelling workflow across different spatial extents and resolution strategies. Independent regional replication of the risk formulation provides a subsequent validation opportunity, as discussed in Section 6.5.4. Similarly, Experiment #2 demonstrates that the risk calculation can be extended to additional locations, times, and aircraft pairs, while large-scale computational scalability is evaluated more directly in Experiment #1.

Beyond these individual requirements, the results demonstrate a common location-based behaviour across all three risk metrics: values remain tied to explicit DGGS locations and times even as the source entities and contributing hazard types change. This is the central property required by Research Question #2 because it enables different analytical products to be derived, interpreted, and combined without replacing the underlying spatial and temporal reference of the airspace model.

6.5.3 Scope and future extensions of the proposed risk formulation

Within Experiment #2, the proposed risk formulation comprises three linked components: a location-based weather risk metric, a location-based MAC risk metric, and a combined weather–MAC risk indicator, all evaluated at common DGGS locations and time steps. These components provide the specific risk-analysis case study used to evaluate the location-based metric computation capability of the thesis methodology.

The MAC risk metric quantifies local aircraft convergence relative to fixed DGGS locations using aircraft position, altitude, motion, and estimated convergence time. Higher indicator values identify locations where qualifying aircraft pairs are converging more rapidly and with more closely matched estimated arrival times. The metric therefore represents the severity of the local aircraft-convergence condition within the proposed location-based analytical framework.

The weather risk metric quantifies the instantaneous convergence of processed storm cell centroids toward fixed DGGS locations using storm cell position, motion, and estimated convergence time. Higher indicator values identify locations toward which a qualifying storm cell is converging more rapidly. The formulation therefore provides a three-dimensional, time-varying weather risk field derived from the selected NEXRAD-based storm cell representation.

The MAC- and weather risk formulations used in Experiment #2 represent two specific applications of the broader DGGS-based metric computation framework. Future research could extend either formulation by incorporating additional information relevant to the target analysis. MAC-related extensions could incorporate aircraft intent, separation criteria, trajectory uncertainty, aircraft performance, or other operational information, while weather-related extensions could incorporate turbulence, icing, windshear, storm growth and decay, forecast uncertainty, or aircraft-specific weather tolerance. Provided that the required information can be associated with appropriate spatial and temporal references, these additional variables can be incorporated into application-specific calculations while retaining the common DGGS-based analytical structure. These extensions would broaden the risk-analysis applications of the methodology while preserving the same underlying location and time reference.

6.5.4 Modelling assumptions and experimental limitations

The normalization of the weather risk and MAC risk components uses selected convergence-time thresholds. These thresholds define the scaling and temporal sensitivity of the 0–100% indicators and are application-specific modelling parameters. The resulting percentages provide a common normalized scale for comparing the relative intensity of the derived indicators across DGGS locations and time steps within the experiment.

The combined risk formulation in Equation (20) uses a complementary-product combination rule that treats the normalized weather risk and MAC risk components as independent contributions for the purpose of combination. This simplifying modelling assumption provides a transparent method for demonstrating how two independently derived, co-indexed analytical products can be integrated. In operational airspace, weather conditions can influence routing, altitude selection, traffic concentration, controller actions, and aircraft-pair convergence geometry, so alternative dependence structures may be appropriate for other applications. Because the component indicators share a common DGGS location and time reference, future implementations can replace the selected combination rule with a conditional, joint, or empirically calibrated relationship while retaining the underlying analytical structure.

The principal experimental limitation of Experiment #2 is that the risk analysis is validated within a single selected 5 km × 5 km × 5 km LGA terminal-area case study and selected historical interactions. This scope was deliberately selected because the LGA case study provides the evidence required to evaluate the risk formulation within Experiment #2, while Experiment #1 separately demonstrates that the underlying DGGS-based airspace-modelling workflow can be applied across different spatial extents and resolution strategies. Independent replication of the risk formulation in an additional geographic region is therefore reserved as a focused validation step for future research. Such replication would provide additional evidence regarding the generalizability of application-specific parameter choices and risk patterns across airports, traffic conditions, and weather regimes.

6.5.5 Response to Research Question #2 and Hypothesis #2

Research Question #2 asks:

How can a reusable 4D airspace model be used to derive and interpret multiple types of location-based airspace metrics from heterogeneous, time-varying airspace data contained within it?

Experiment #2 provides a specific answer to this question through the risk case study. Heterogeneous storm cell and aircraft observations are first retained within a common 4D airspace representation using DGGS locations and time references. Application-specific calculations are then used to transform selected contents of that model into new quantities associated with the same locations and times. Because the resulting quantities are co-indexed, they can be interpreted independently, compared directly, and combined without reconstructing the underlying airspace representation for each analytical application.

The weather- and MAC risk metrics demonstrate this process using two physically distinct source datasets. Radar-derived storm cell observations are transformed into a location-based weather risk field, while ADS-B aircraft trajectories are independently transformed into a location-based MAC risk field. The resulting metrics show interpretable variation across three-dimensional space and time, consistent with the geometry of their respective source observations. Their common DGGS reference then enables the two fields to be combined into a third analytical product.

Experiment #2 therefore demonstrates how a reusable 4D airspace model can support multiple derived location-based quantities by maintaining a common spatial and temporal reference between the source observations and the application-specific calculations performed on them. The result demonstrates this broader analytical mechanism through weather risk, aircraft-proximity risk, and their combination as a controlled multi-source case study, while application of the same mechanism to additional airspace metrics remains an opportunity for subsequent research.

Hypothesis #2 states:

A reusable DGGS-based 4D airspace model can support multiple co-indexed location-based analytical products. A layered risk analysis case study using aircraft proximity risk, hazardous weather risk, and their combined effects can demonstrate this capability.

The results support Hypothesis #2 for the intended experimental application. The weather risk indicator varies in a physically interpretable way as storm cell entities move relative to selected DGGS locations. The MAC risk indicator identifies locations and times when aircraft pairs exhibit increasingly rapid convergence to a common location. The combined field identifies conditions in which weather and aircraft contributions occur independently or simultaneously at the same DGGS locations and times. The use of three-dimensional maps, 2D horizontal risk maps, and time-series results further demonstrates that these patterns can be interpreted across horizontal position, altitude, and time.

This support is subject to the scope established above. The resulting values are normalized analytical indicators rather than calibrated operational event probabilities, and the combined formulation includes an independence assumption introduced for analytical demonstration. Within these limitations, Experiment #2 provides evidence for the prediction made by Hypothesis #2 and answers the broader Research Question #2 by demonstrating a concrete way to transform reusable heterogeneous 4D airspace data into multiple co-indexed analytical products.

Chapter 7: Behaviour-based aircraft categorization using machine learning

This chapter presents Experiment #3, which evaluates the **unsupervised machine learning branch of the methodology** developed in Chapter 4 in order to determine whether trajectory-level outputs from the reusable 4D airspace model can be used to identify recurring, operationally meaningful patterns in a data-driven manner when the relevant groupings are not defined in advance. Behaviour-based aircraft categorization serves as the case study for evaluating this capability. Whereas Experiment #2 examined a location-based analytical use of the airspace model by deriving metrics at common three-dimensional locations and times, Experiment #3 evaluates a fundamentally different downstream use by analysing complete aircraft trajectories extracted from the same modelling workflow.

The experiment proposed in Section 3.3 is to use unsupervised clustering to identify recurring structure directly from the observed trajectory data. Complete aircraft trajectories are transformed into a common representation and grouped based on similarities in their spatial and temporal evolution. The results are validated using operational knowledge and independent aircraft-reference information to characterize, interpret, and compare the resulting groups.

The principal validation presented in this chapter concerns whether the selected KMeans implementation identifies recurring operational structure from the trajectory data, how that structure changes with category granularity, and how the resulting groups relate to independent aircraft-reference information. The behaviour-based aircraft categorization application builds on the work presented in [1] and [4].

Figure 86 shows the relationship between Experiment #3 and the overall methodology developed in Chapter 4.

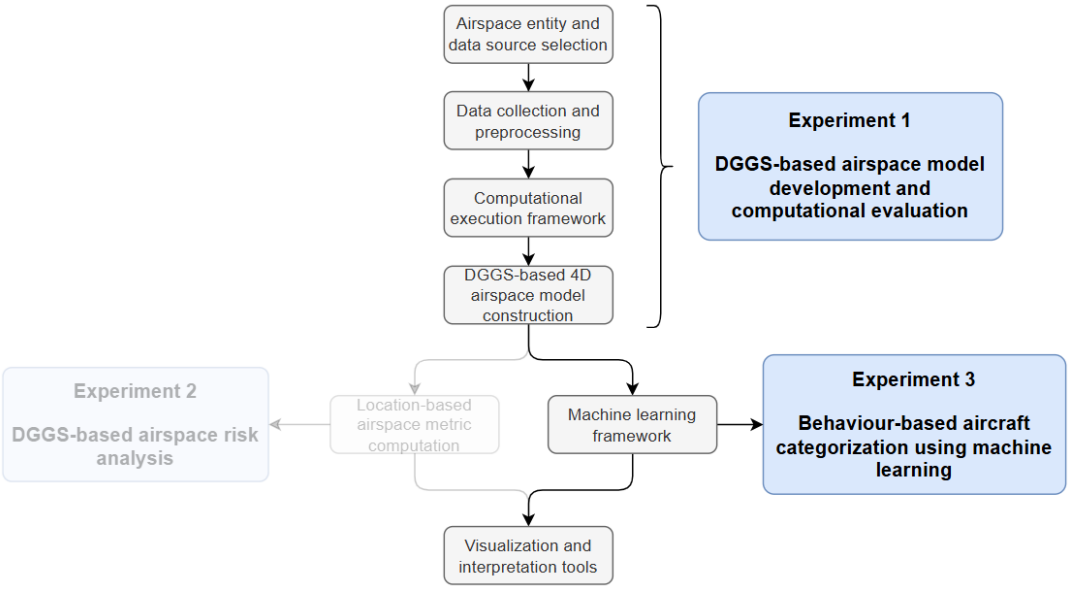


Figure 86: Relationship between methodology and Experiment #3

The experiment addresses Research Question #3, which asks:

How can trajectory-level information from a reusable 4D airspace model be used to identify recurring operationally meaningful patterns in a data-driven manner and support their interpretation when the relevant groupings are not defined in advance?

Hypothesis #3 is proposed to address this question:

Unsupervised clustering applied to trajectory-level outputs from a reusable 4D airspace model can identify recurring aircraft operational patterns directly from observed trajectory structure without predefined behavioural labels, and the resulting data-derived groups can subsequently be interpreted as operationally meaningful behaviour categories using post-clustering spatial, kinematic, and domain evidence.

The chapter is organized as follows. Section 7.1 defines the experimental objective and workflow. Section 7.2 describes the trajectory dataset and its fixed-length clustering representation. Section 7.3 presents the behaviour-based categorization results obtained using the selected KMeans implementation. Section 7.4 evaluates how the identified trajectory structure changes with category granularity. Section 7.5 compares the resulting trajectory groups with independent aircraft-reference categories. Section 7.6 interprets the combined results and evaluates them against Research Question #3 and Hypothesis #3.

7.1 Experimental objective and workflow

Experiment #3 operationalizes Hypothesis #3 through the workflow shown in Figure 87. Trajectory-level outputs are first extracted from Airspace Model #1 and transformed into a common fixed-length representation. The selected KMeans implementation then assigns each trajectory to a data-driven cluster. The resulting trajectory groups are analyzed using three complementary forms of post-clustering evidence: spatial and kinematic characterization, category-granularity analysis, and comparison with independent aircraft-reference information. These analyses collectively support the operational interpretation and validation of the resulting behaviour-based aircraft categories.

Throughout this chapter, a **cluster** refers to a trajectory group generated directly by the clustering algorithm, whereas a **behaviour category** refers to the operational interpretation assigned to that group. This distinction separates the computational grouping stage from the domain-specific interpretation stage. The clustering representation determines the initial group assignments, while spatial and kinematic variables, operational context, and independent aircraft-reference information are then used to characterize and interpret those groups.

Behaviour-based categorization is intended to capture a different property than conventional aircraft classification. Conventional classifications describe aircraft using attributes such as physical configuration, aircraft type, or airworthiness category, whereas the proposed approach groups trajectories according to recurring patterns in how aircraft move through the observed airspace. Section 7.5 then compares these two forms of categorization using independent aircraft-reference information.

KMeans is used as the clustering algorithm for the remainder of Experiment #3. A supporting comparison of DBSCAN, HDBSCAN, KMeans, and GMM was conducted using the same trajectory-analysis application and is presented in Appendix C and [3]. That analysis identified KMeans as the most suitable tested clustering algorithm for the present aircraft categorization application, including complete trajectory assignment and the interpretability of the resulting groups.

The primary output of the clustering stage is a trajectory-level cluster assignment for each input trajectory. The Experiment #3 results are evaluated against the four success criteria established in Section 3.3:

1. the identification of recurring trajectory groups directly from the airspace-model outputs;
2. distinct spatial and kinematic characteristics supporting operational interpretation;
3. coherent finer subpatterns when category granularity is increased; and
4. post-clustering comparison with independent aircraft-reference information.

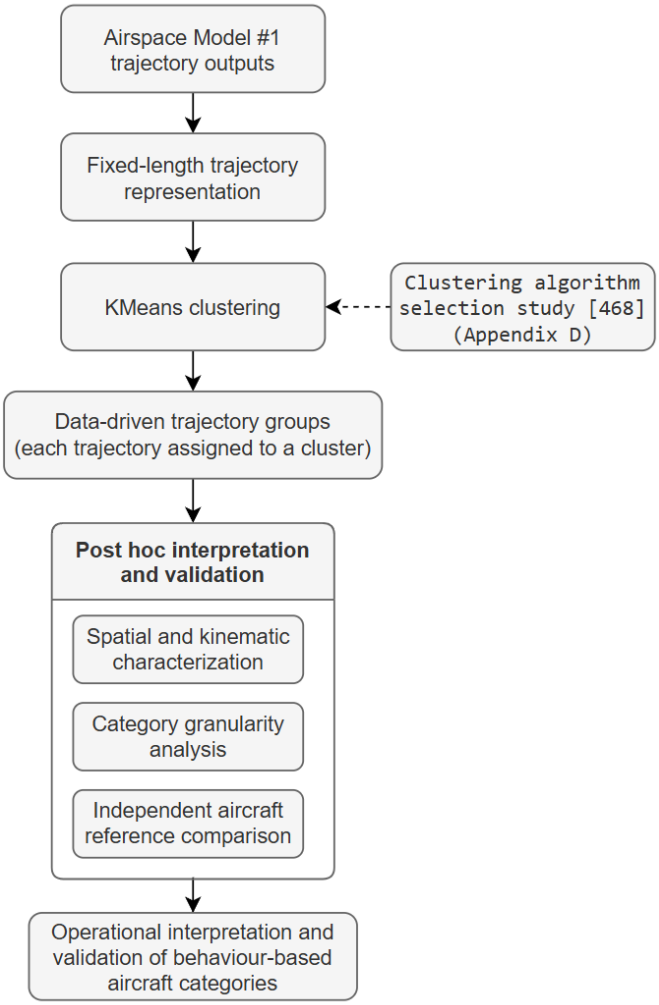


Figure 87: Experiment #3 workflow for data-driven identification and interpretation of aircraft behaviour categories

7.2 Trajectory dataset and clustering representation

This section describes the trajectory-level airspace-model outputs used in Experiment #3 and how they are transformed into the common fixed-length representation supplied to the clustering algorithm.

7.2.1 Input dataset

The input to Experiment #3 consists of trajectory-level ADS-B data extracted from Airspace Model #1. The dataset covers the five-hour observation interval in the LaGuardia-centred terminal-area model developed in Experiment #1. After preprocessing and trajectory segmentation, the clustering dataset comprises 346 trajectory units. Each trajectory unit represents a temporally continuous sequence of observations from a single aircraft and constitutes one clustering sample.

Trajectory segmentation can produce multiple trajectory units from the same aircraft when discontinuities occur in the underlying observations. The 346 trajectory units used for clustering therefore correspond to 343 unique aircraft in the independent FAA/OpenSky comparison presented in Section 7.5.

Figure 88 and Figure 89 present two- and three-dimensional views of the trajectory dataset used in the clustering workflow. At this stage, their purpose is to illustrate the spatial structure of the input data before any cluster assignments or operational interpretations are introduced.

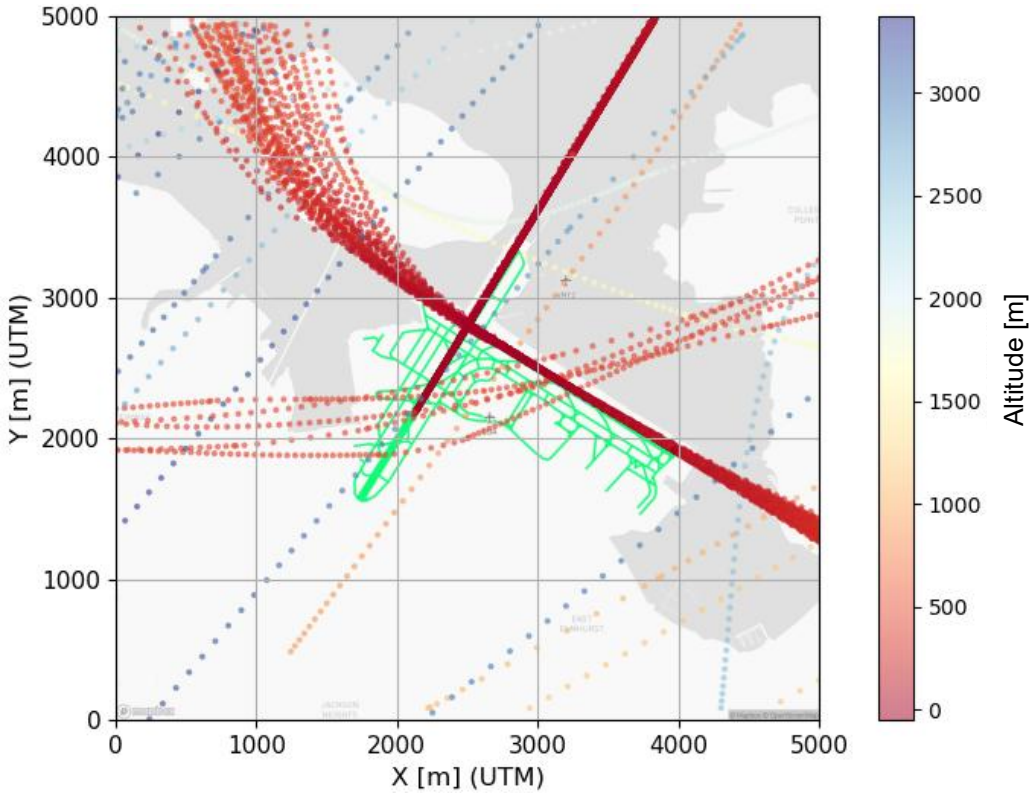


Figure 88: 2D view of the trajectory-level clustering input dataset

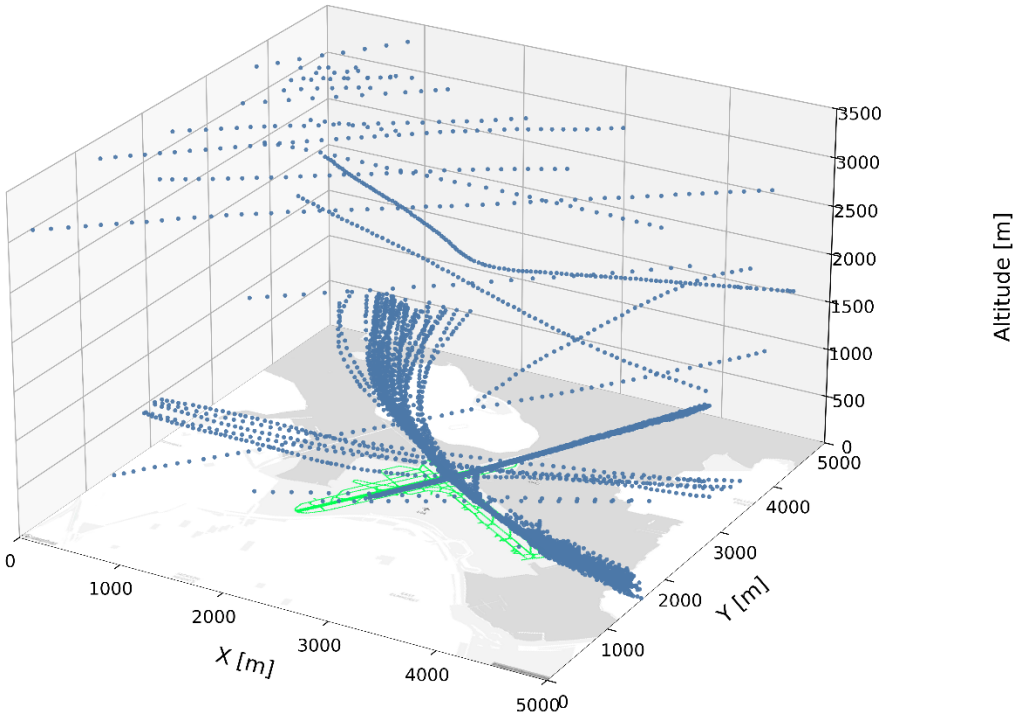


Figure 89: 3D view of the trajectory-level clustering input dataset

7.2.2 Fixed-length trajectory representation

The trajectory observations are processed using the data-collection and preprocessing workflow described in Chapters 4 and 5. Observations for each trajectory unit are ordered chronologically, duplicate timestamps are removed, and trajectories of varying duration are represented on a common normalized time axis ranging from 0 at the start of the trajectory to 1 at its end.

A common fixed-length representation is required because the trajectory units contain different numbers of observations and span different durations. KMeans and the candidate clustering algorithms evaluated in Appendix C operate on samples represented within a common feature space and therefore require each trajectory to be expressed using the same number and ordering of input variables. Normalizing trajectory time and interpolating each trajectory to a common sequence length allows complete trajectory shapes to be compared on a consistent basis while retaining their relative spatial evolution from beginning to end.

Longitude, latitude, and geometric altitude are interpolated at 64 equally spaced normalized-time positions. The resulting horizontal-position and altitude sequences are concatenated into a fixed-length trajectory-level feature vector, along with trajectory duration and the number of original observations. The complete feature matrix is then standardized before KMeans clustering. Normalized trajectory time serves as the common alignment coordinate rather than as an additional spatial feature.

Figure 90 illustrates the aligned trajectory representation by plotting the interpolated horizontal-position and altitude sequences against normalized trajectory time. Each trajectory therefore contains corresponding spatial samples at the same relative positions between its beginning and end, despite differences in its original duration or number of observations. The clustering input consists of these fixed-length spatial sequences together with trajectory duration and observation count. Heading, vertical rate, and aircraft velocity are retained separately for post-clustering characterization in Section 7.3.

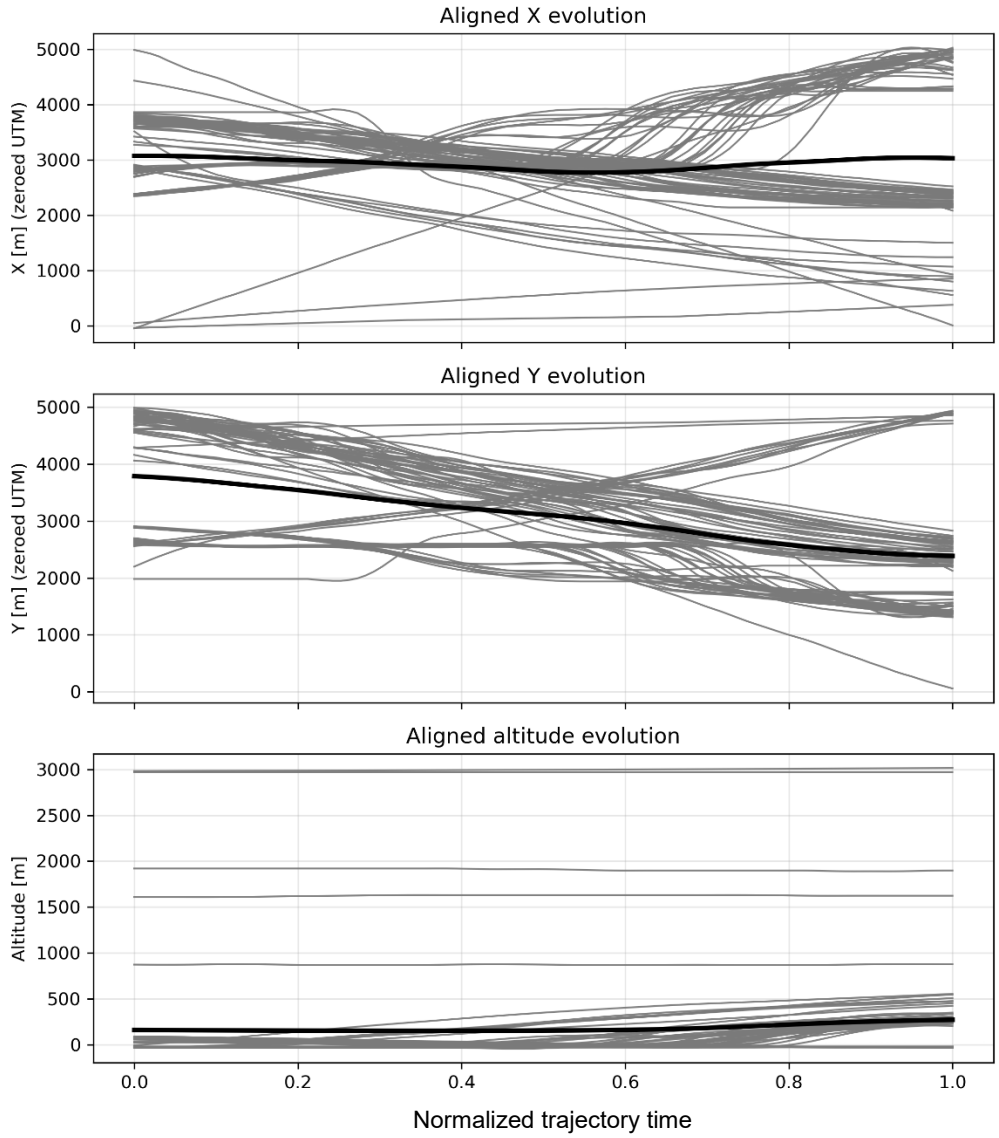


Figure 90: Aligned parameters over normalized time

7.3 Behaviour-based aircraft categorization results

The selected KMeans implementation was applied to the 346 trajectory units with K=5 as the principal category granularity. Each trajectory was assigned to one of five clusters using only the clustering representation described in Section 7.2. The resulting groups are identified numerically as clusters 0 through 4 throughout this results section; their operational interpretation is presented separately in Section 7.6.1.

The results are characterized using complementary views of trajectory geometry and aircraft motion. A common cluster identifier is retained across the 2D trajectory map, cluster-size distribution, altitude, heading, vertical rate, velocity, and standardized-variable figures, enabling consistent comparison of the same five trajectory groups.

Figure 91 shows the 2D distribution of the five KMeans clusters. Clusters 0, 1, and 3 occupy three dense and spatially distinct trajectory corridors within the terminal-area region, while clusters 2 and 4 contain smaller sets of trajectories following different paths across the model domain. The spatial grouping established in this figure provides the common cluster reference used by the subsequent parameter-based results.

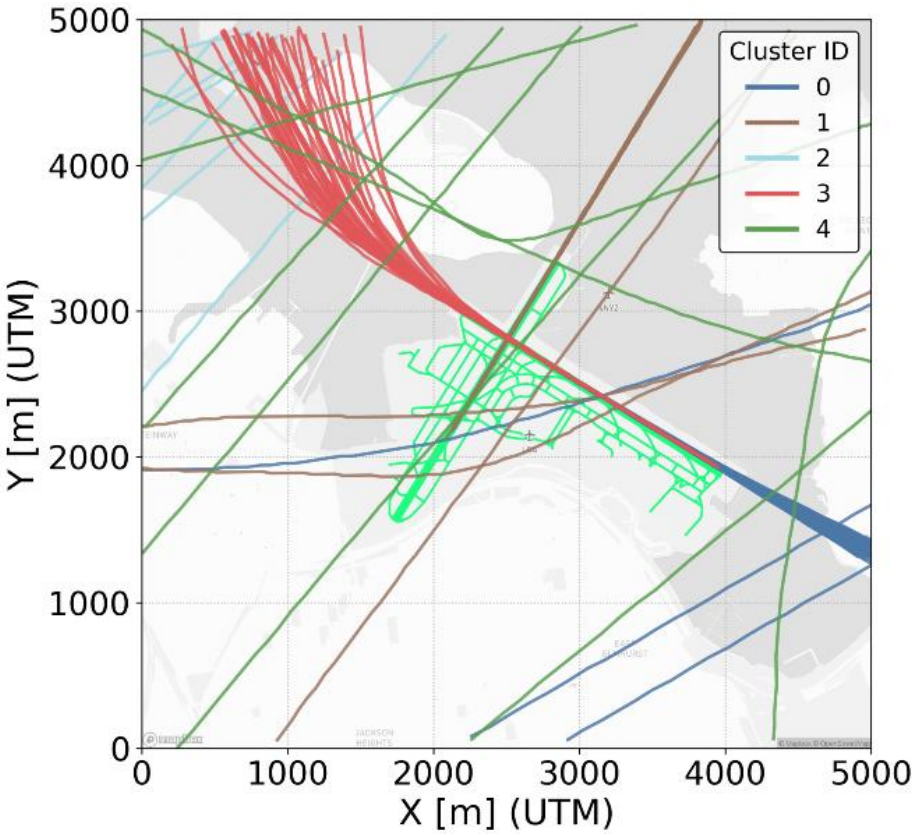
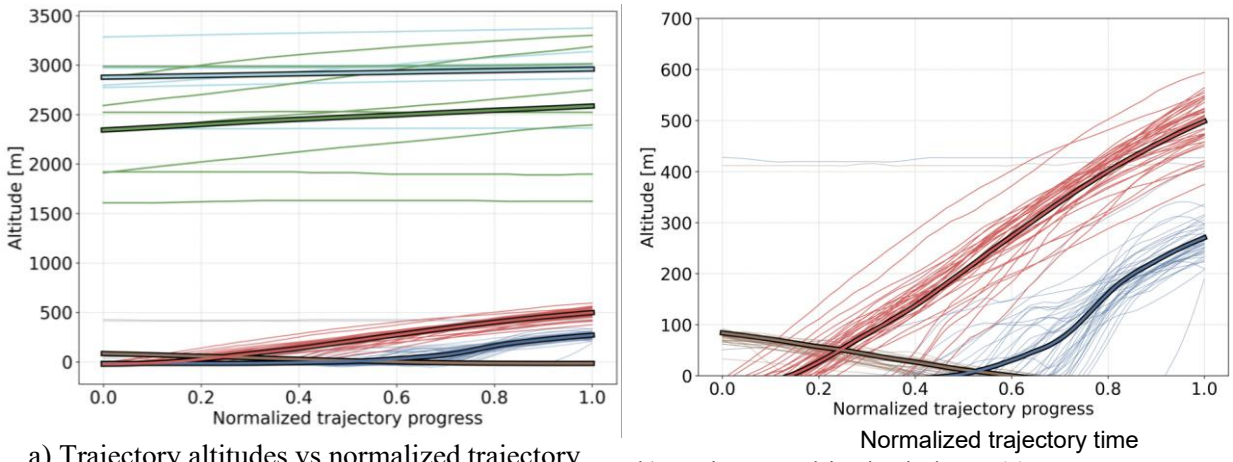


Figure 91: 2D view of the K=5 KMeans trajectory clusters

Figure 92 shows altitude as a function of normalized trajectory time for the five clusters shown in Figure 91. Clusters 0, 1, and 3 are concentrated predominantly below 600–700 m, whereas clusters 2 and 4 contain trajectories extending from 1,600 m to 3,300 m. Within the lower-altitude groups, Cluster 1 generally decreases in altitude over normalized trajectory time, whereas Clusters 0 and 3 generally increase in altitude.



a) Trajectory altitudes vs normalized trajectory time

b) Trajectory altitudes below 700 m

Figure 92: Altitude over normalized time for each trajectory coloured by cluster ID

Figure 93 shows the number of trajectories assigned to each of the five clusters. Cluster 1 is the largest, with 168 trajectories (48.6%), followed by Cluster 0 with 123 trajectories (35.5%). Cluster 3 contains 38 trajectories (11.0%), while Clusters 2 and 4 contain 9 (2.6%) and 8 (2.3%) trajectories, respectively. The five clusters therefore differ substantially in the number of trajectories assigned to them, with Clusters 0 and 1 together accounting for 291 of the 346 trajectory units (84.1%).

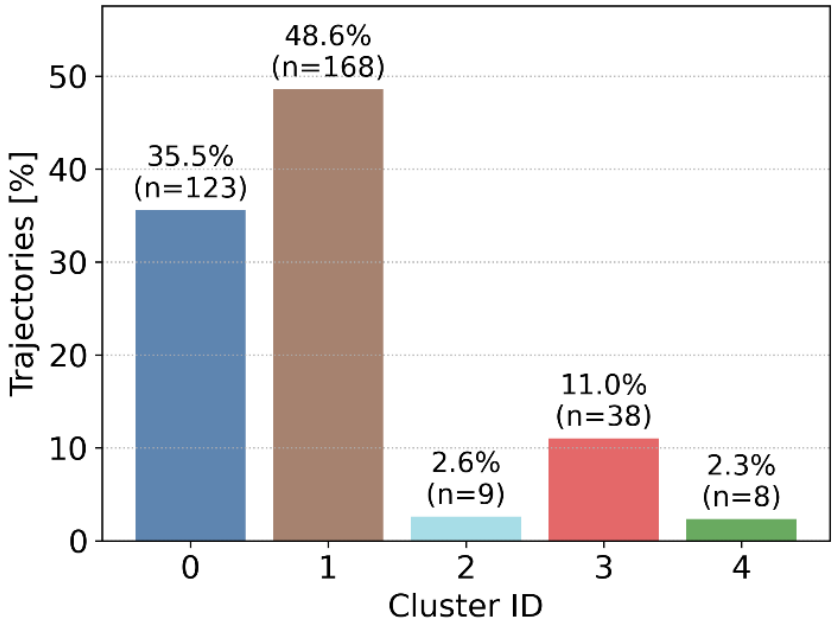


Figure 93: Cluster-size distribution

Figure 94 shows the heading profiles for the clustered trajectories shown in Figure 91. Cluster 0 remains concentrated near 130° for much of the normalized trajectory time, while Cluster 1 remains concentrated near 220°. Cluster 3 initially occupies a heading near 310° and then shifts toward 350°. Clusters 2 and 4 follow heading evolutions distinct from those of the three larger low-altitude clusters.

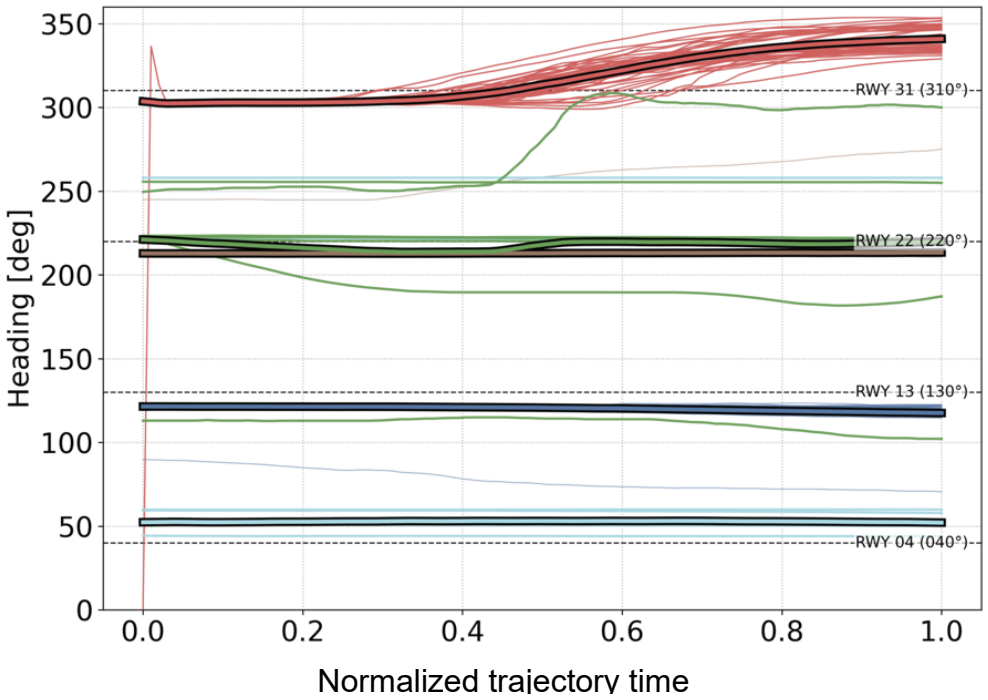


Figure 94: Heading profiles over normalized trajectory time

Figure 95 shows vertical rate as a function of normalized trajectory time. Clusters 0 and 3 exhibit predominantly positive vertical rates, although the timing and magnitude of the increase differ between the two groups. Cluster 1 exhibits predominantly negative vertical rates. Cluster 2 remains comparatively close to zero vertical rate for most of the normalized trajectory, while Cluster 4 exhibits a generally positive vertical rate of lower magnitude than the principal increasing-altitude groups.

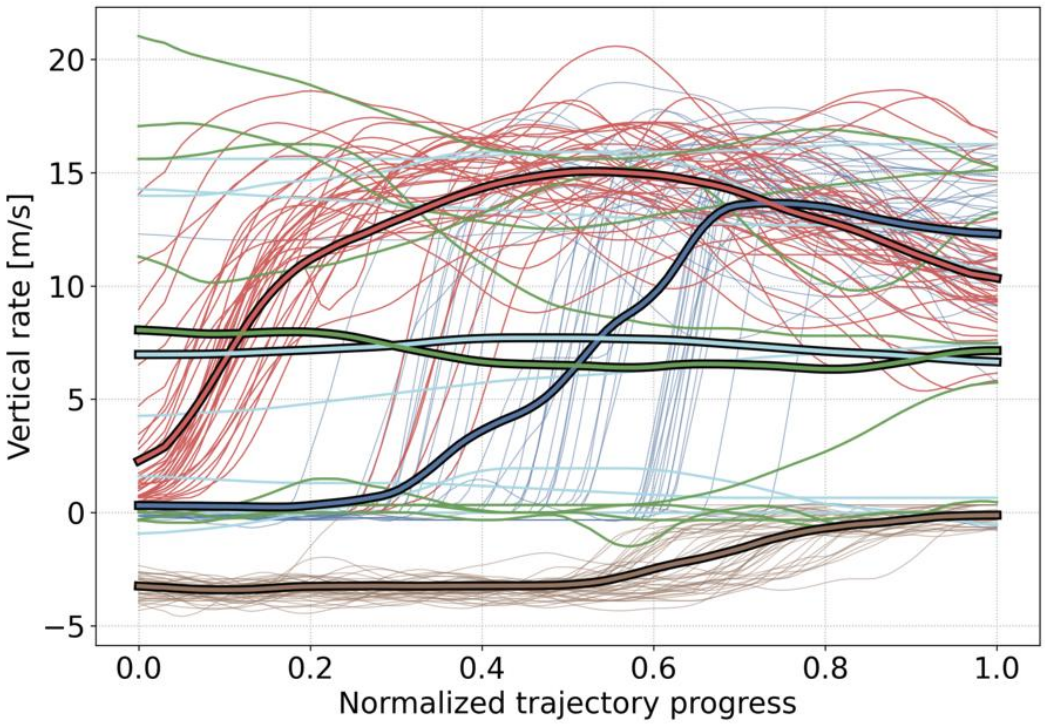


Figure 95: Vertical rate over normalized time

Figure 96 shows aircraft velocity as a function of normalized trajectory time. Cluster 2 contains the highest-velocity trajectories, remaining within the 385–395 kt range for much of normalized trajectory time. Cluster 4 is also characterized by comparatively high velocities of 310–320 kt. Clusters 0 and 3 occupy lower velocity ranges and generally increase in velocity over normalized trajectory time, whereas Cluster 1 decreases from 140–150 kt to 90–100 kt.

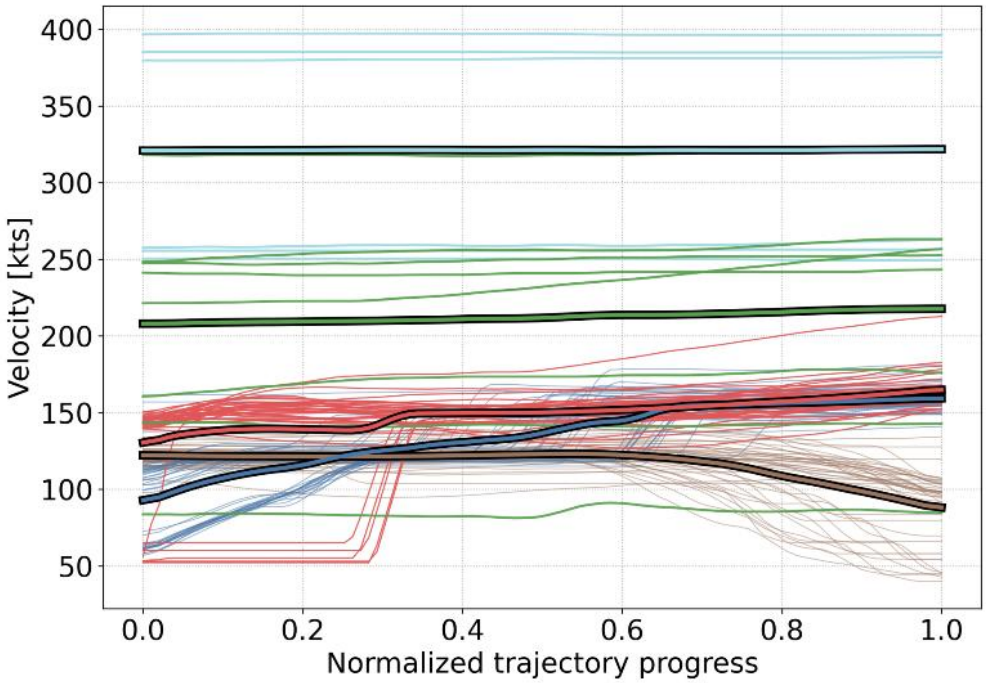


Figure 96: Velocity over normalized time

Figure 97 summarizes selected trajectory-level characteristics of the five clusters using standardized values relative to the dataset mean. Cluster 2 has the highest standardized mean altitude (+4.91) and mean velocity (+4.60), while Cluster 4 also has a high mean altitude (+4.11) and an above-average mean velocity (+1.83). Cluster 3 shows comparatively high altitude variation (+2.02), vertical rate (+1.80), and heading variation (+2.58). Cluster 1 has below-average mean velocity (−0.61) and vertical rate (−0.76). The heatmap therefore provides a compact quantitative summary of the differences among the same five groups shown in the trajectory and time-series results.

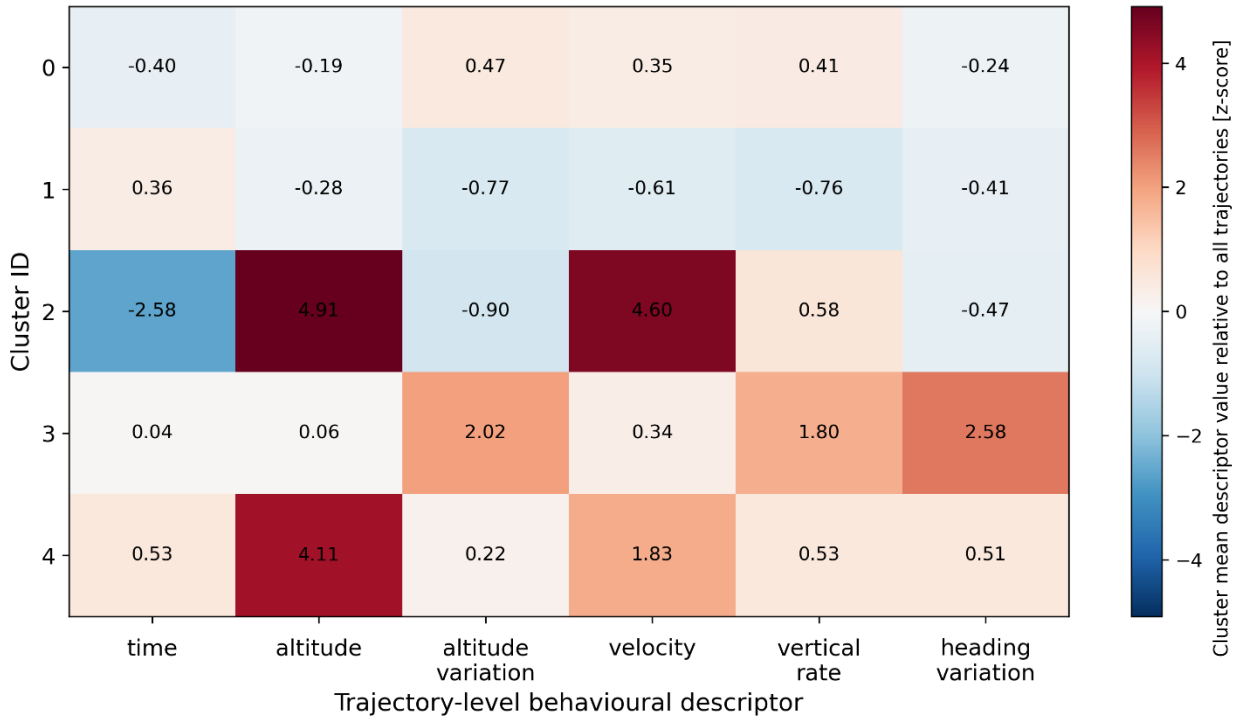


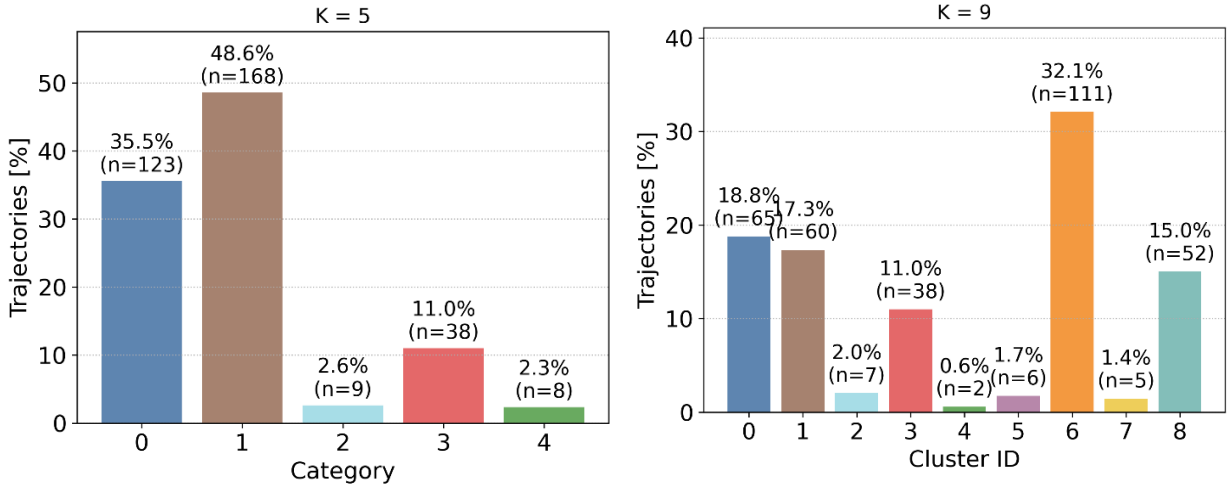
Figure 97: Normalized aircraft behaviour variable heatmap for the K=5 KMeans model

Taken together, the K=5 results show that the five algorithm-generated trajectory groups differ across multiple characteristics, including 2D trajectory geometry, altitude evolution, heading, vertical rate, velocity, and standardized trajectory-level variables. Section 7.6.1 uses these complementary differences to interpret the operational meaning of the identified groups.

7.4 Effect of category granularity

For KMeans, the specified number of clusters K determines the level of detail at which recurring trajectory structure is represented. In this experiment, this is referred to as category granularity: a lower value of K produces fewer, broader trajectory groups, whereas a higher value allows those groups to be subdivided into finer trajectory patterns. The $K = 5$ configuration is retained as the principal high-level categorization, while $K = 9$ is used to evaluate how the same dataset is partitioned at a finer granularity.

Figure 98 compares the cluster-size distributions for $K = 5$ and $K = 9$. At $K = 5$, the largest cluster contains 168 trajectories (48.6%), and the two largest clusters together account for 84.1% of the dataset. At $K = 9$, the largest cluster contains 111 trajectories (32.1%), with the other groups containing 65 (18.8%), 60 (17.3%), 52 (15.0%), and 38 (11.0%) trajectories, together with several smaller groups containing between 2 and 7 trajectories. The finer configuration therefore distributes the trajectories across a larger number of groups rather than concentrating most of the dataset within the two dominant clusters.

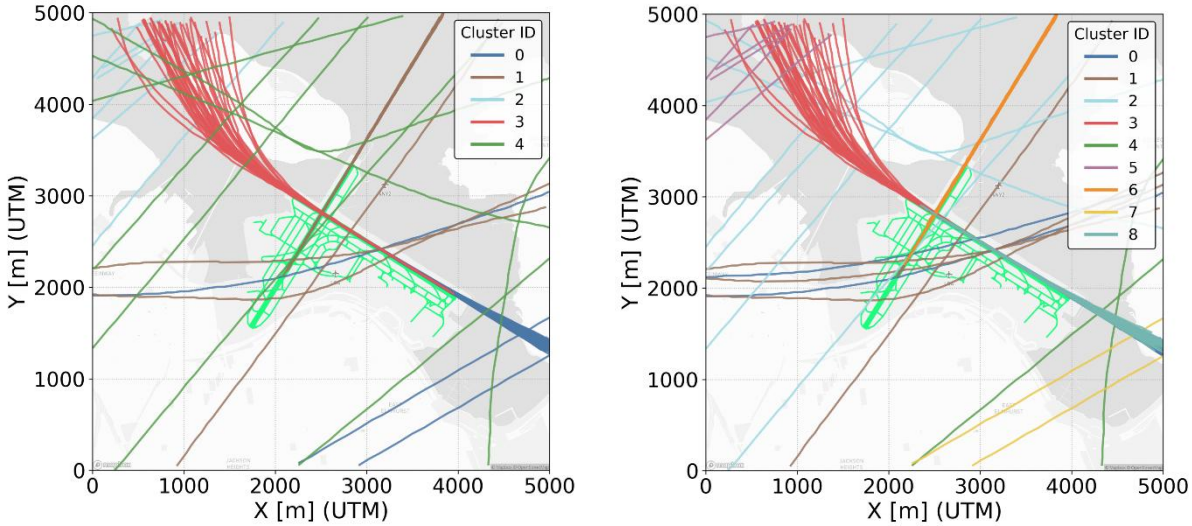


a) K=5 model results

b) K=9 model results

Figure 98: Cluster distribution comparison for the k=5 and k=9 KMeans models

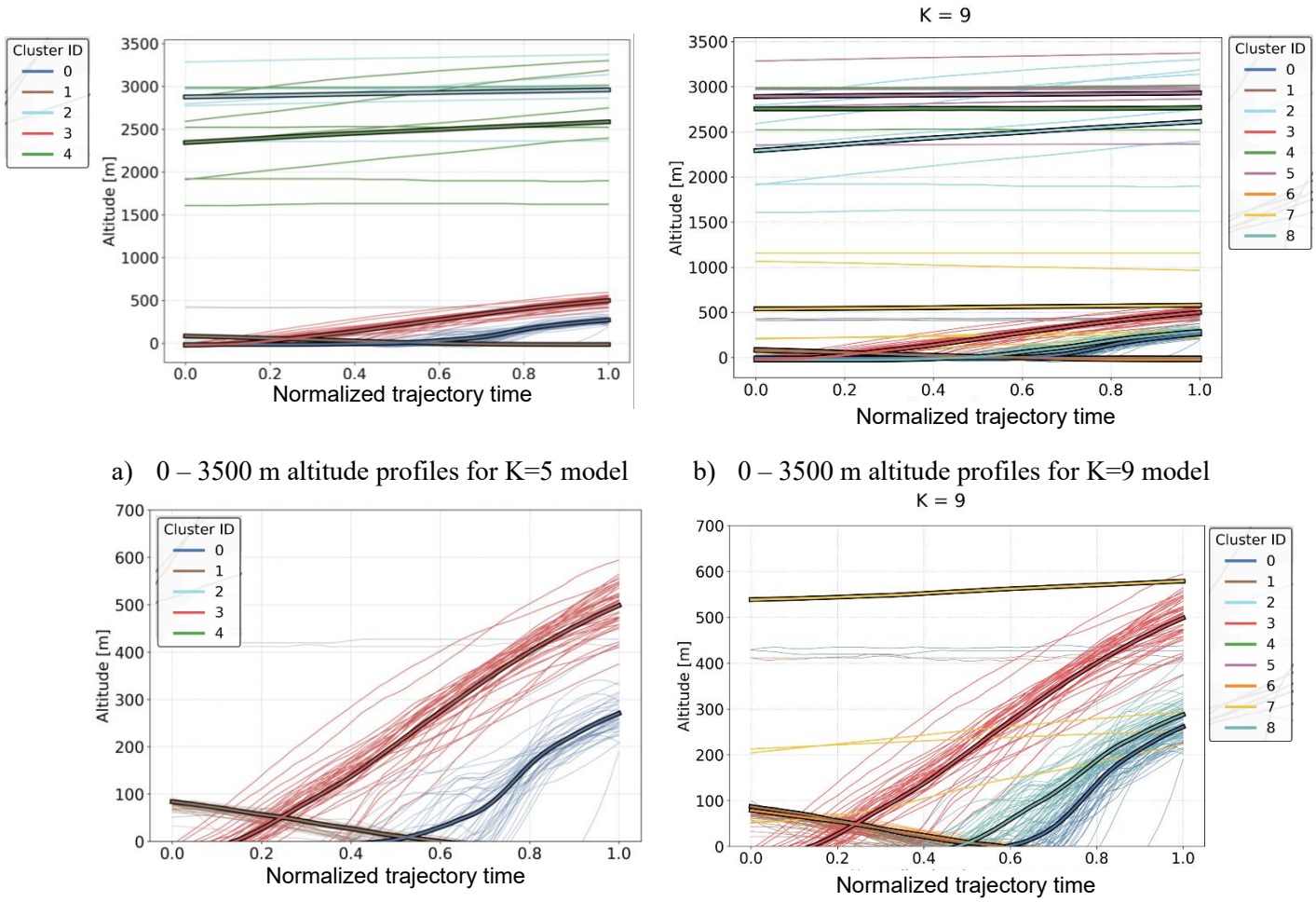
Figure 99 compares the two-dimensional trajectory groupings obtained at $K = 5$ and $K = 9$. The broader spatial structure identified at $K = 5$ remains recognizable, while several groups are subdivided at the finer granularity. Two particularly clear examples are visible in the figure: trajectories assigned to Cluster 0 for $K = 5$ are divided primarily between clusters 0 and 8 for $K = 9$, while trajectories assigned to Cluster 1 for $K = 5$ are divided primarily between clusters 1 and 6 for $K = 9$. The finer configuration therefore resolves additional structure within trajectory families that are represented as single groups in the $K = 5$ solution.



a) K=5 model results b) K=9 model results

Figure 99: 2D map of clustered trajectories for a) k=5 and b) k=9 KMeans models

Figure 100 shows that the subdivisions visible in the 2D trajectories are also associated with differences in altitude evolution. Several broad $K = 5$ groups are further divided into $K = 9$ groups with more specific altitude histories. This is evident across the full altitude range and within trajectories below 700 m, where the finer configuration reveals additional vertical movement patterns within trajectory groups that are combined at $K = 5$.



a) 0 – 3500 m altitude profiles for K=5 model

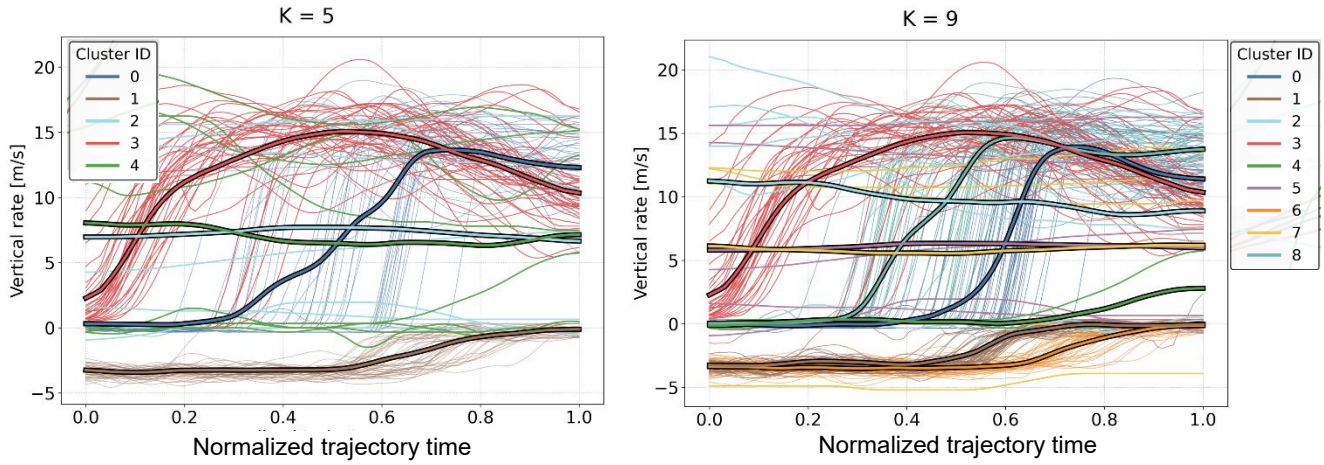
b) 0 – 3500 m altitude profiles for K=9 model

c) Below 700 m altitude profiles for K=5 model

d) Below 700 m altitude profiles for K=9 model

Figure 100: Altitude over time comparison for the k=5 and k=9 KMeans models

Figure 101 shows the corresponding vertical-rate comparison. At $K = 5$, broad positive, negative, and near-zero vertical-rate patterns are further subdivided at $K = 9$ into groups with distinct timing and magnitude of vertical motion (i.e., cluster 0 from $K = 5$ is split into clusters 0 and 8 at $K = 9$, and cluster 1 is split into clusters 1 and 6 at $K = 9$). The finer configuration therefore resolves additional within-group variation while retaining the principal vertical-motion structure identified at $K = 5$.



a) K=5 model results

b) K=9 model results

Figure 101: Vertical rate over time comparison for the k=5 and k=9 KMeans models

Taken together, the $K = 5$ and $K = 9$ results show that increasing K yields a finer representation of trajectory structure by subdividing broader groups into more detailed spatial and kinematic patterns. The $K = 5$ result provides the compact categorization used in the principal aircraft-behaviour analysis, while the $K = 9$ result shows that the same workflow can capture operational structure at a finer level of granularity. The operational meaning of the identified subdivisions is examined in Section 7.6.2.

7.5 Validation through comparison with independent aircraft-reference categories

The final post-clustering analysis compares the $K = 5$ trajectory groups with independent aircraft-reference information. Aircraft attributes obtained from the FAA Aircraft Registration Database [464] and the OpenSky aircraft database [59] are associated with the clustered trajectories after the KMeans assignments have been generated. The resulting comparison characterizes the relationship between the data-driven trajectory groups and conventional technical aircraft categories.

The independent reference information includes airworthiness-related categories derived from aircraft configuration and registration information. These categories are used only for post-clustering comparison and therefore remain independent of the trajectory representation used to generate the KMeans groups. The independent aircraft-reference information is obtained by combining the OpenSky aircraft database [59] and the FAA Aircraft Registration Database [464]. Aircraft are matched using their ICAO 24-bit identifiers, with the OpenSky database serving as the primary source and FAA registration records providing additional information for aircraft without an OpenSky match. The OpenSky database aggregates information from official aircraft registries and community-contributed sources, providing broad aircraft coverage, while the FAA database provides official registration and aircraft reference information for U.S.-registered aircraft. The resulting metadata are used to assign conventional aircraft categories for post-clustering interpretation.

This combined approach provides category information for 342 of the 343 unique aircraft (99.7%) included in the independent comparison, with only one aircraft (0.3%) remaining unknown or uncategorized.

Figure 102 shows the airworthiness-category composition of the 343 unique aircraft included in the external comparison. Medium fixed-wing aircraft account for 306 aircraft (89.2%), large fixed-wing aircraft for 28 (8.2%), rotorcraft for 5 (1.5%), small fixed-wing aircraft for 3 (0.9%), and the unknown or uncategorized group for 1 aircraft (0.3%). The independent comparison dataset is therefore strongly dominated by medium fixed-wing aircraft.

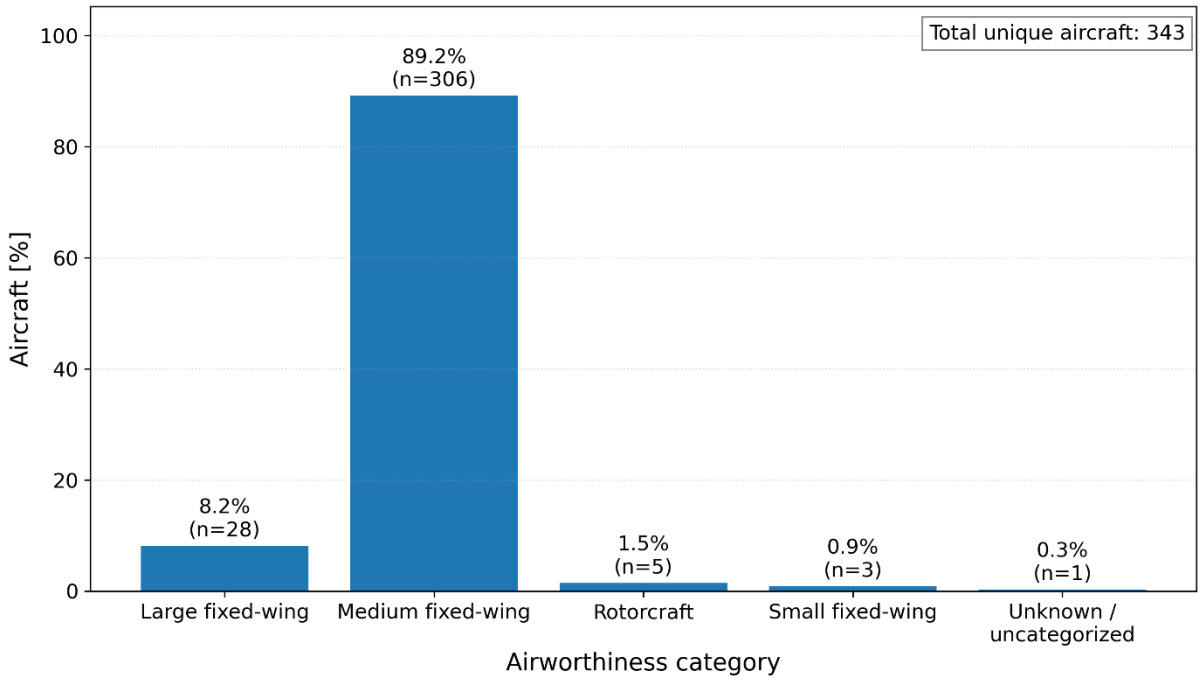


Figure 102: Airworthiness category distribution of the input ADS-B dataset

Figure 103 shows the within-cluster distribution of the independent airworthiness categories across the five $K = 5$ trajectory groups. The comparison indicates how closely each trajectory-derived cluster aligns with, or cuts across, conventional technical aircraft categories. A cluster dominated by a single conventional category yet exhibiting operational characteristics distinct from other clusters with the same conventional composition would indicate that the trajectory-based grouping captures information beyond aircraft technical classification alone.

Medium fixed-wing aircraft are the dominant reference category across all clusters. They account for 93.4% of Cluster 0, 88.7% of Cluster 1, and 92.1% of Cluster 3. Cluster 2 comprises 57.1% medium fixed-wing and 42.9% large fixed-wing aircraft, while Cluster 4 comprises 50.0% medium fixed-wing, 25.0% small fixed-wing, 12.5% large fixed-wing, and 12.5% unknown or uncategorized aircraft.

The comparison therefore shows that several independently generated trajectory clusters have similar dominant conventional aircraft-category compositions yet remain distinct trajectory groups. The operational significance of this relationship is examined in Section 7.6.3.

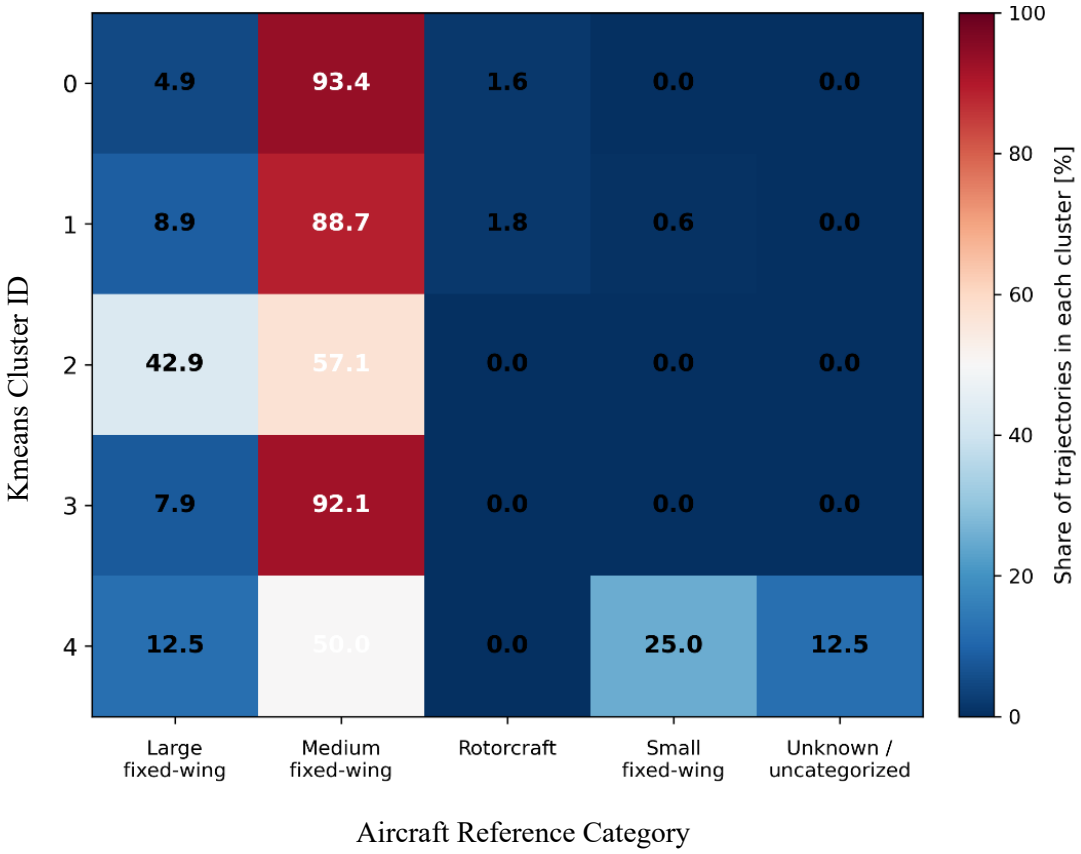


Figure 103: Within-cluster distribution of independent aircraft-reference categories for the K=5 KMeans trajectory groups

7.6 Experiment #3 discussion

Sections 7.3–7.5 presented the results of Experiment #3, separating the algorithm-generated trajectory groups from their subsequent operational interpretation. This section integrates those results to interpret the behaviour represented by the $K = 5$ trajectory groups, examine the operational meaning of category granularity, evaluate the relationship between the resulting behaviour-based categories and the independent aircraft-reference categories, and determine how the combined evidence addresses Research Question #3 and Hypothesis #3.

7.6.1 Operational interpretation of the identified behaviour categories

The $K = 5$ KMeans configuration produces five trajectory groups using the fixed-length representation described in Section 7.2. Their operational meaning is assigned after clustering by relating the generated groups to the complementary spatial and kinematic results presented in Section 7.3. This distinction is important to the data-driven formulation of Experiment #3: cluster membership is determined from the selected trajectory representation without predefined behavioural labels, while domain knowledge is subsequently used to interpret the resulting structure.

Clusters 0, 1, and 3 represent three distinct terminal-area behaviours. Cluster 0 follows the trajectory corridor aligned with the approximate 130° heading of runway 13 and is characterized by increasing altitude and velocity over time, with a predominantly positive vertical rate. Together, these characteristics support its interpretation as a runway 13 departure family. Cluster 1 follows the approximate 220° heading of

runway 22 and exhibits decreasing altitude, a predominantly negative vertical rate, and decreasing velocity, supporting its interpretation as a runway 22 arrival family. Cluster 3 initially follows the approximate 310° heading of runway 31 before turning toward approximately 350°. Its increasing altitude, positive vertical rate, and comparatively large heading variation support its interpretation as a runway 31 departure-and-turn behaviour.

Clusters 2 and 4 represent less frequent, higher-altitude behaviours distinct from the three principal terminal-area groups. Cluster 2 contains the highest-altitude and highest-velocity trajectories and remains close to level flight, consistent with higher-altitude transit or overflight over the LGA airport region. Cluster 4 also occupies substantially higher altitudes than the terminal groups but is characterized by lower velocity than Cluster 2 and a modest positive vertical rate, indicating a separate higher-altitude movement pattern with a climbing component. Although these groups contain substantially fewer trajectories than clusters 0, 1, and 3, their separation demonstrates that the selected clustering implementation retains lower-frequency trajectory structure in addition to the dominant terminal flows.

The combined spatial and kinematic evidence therefore supports the interpretation of the five algorithm-generated groups as distinct aircraft behaviour categories. In particular, heading, vertical rate, and velocity were reserved for post-clustering characterization rather than used to generate the cluster assignments. Their subsequent agreement with the trajectory geometry and altitude structure provides complementary evidence that the generated groups represent meaningful differences in how aircraft operate within the observed airspace. These results satisfy the first two Experiment #3 success criteria: recurring groups are identified without predefined behavioural category labels, and those groups exhibit distinct characteristics that support operational interpretation.

7.6.2 Operational meaning of category granularity

Category granularity represents the level of operational detail at which the trajectory structure is described. The $K = 5$ configuration provides a compact, high-level representation of the principal behaviour families, whereas the $K = 9$ configuration resolves additional structure within several of those same families. The choice of K can therefore be interpreted as an application-dependent choice of behavioural resolution.

The relationship between the $K = 5$ and $K = 9$ results is particularly clear for the two dominant terminal groups. Trajectories assigned to Cluster 0 at $K = 5$ are split between clusters 0 and 8 at $K = 9$, while those assigned to Cluster 1 at $K = 5$ are split between clusters 1 and 6 at $K = 9$. In both cases, the broader runway-aligned spatial structure remains recognizable, while the finer groups merging at $K = 9$ exhibit distinct altitude and vertical-rate evolutions. Operationally, this shows that a broad departure or arrival family represented as a single category at $K = 5$ can be represented at higher granularity as subgroups with different vertical-development profiles, including differences in the timing and magnitude of climb or descent.

An important implication of this comparison is that the operational meaning identified by the clustering analysis does not depend on a single predetermined value of K. Selecting K remains an analyst-defined modelling choice, but its role is to set the level of granularity at which the trajectory structure is represented, rather than to prescribe the operational identity or membership of the resulting categories. The $K = 5$ and $K = 9$ results provide evidence of this distinction: the principal trajectory structure remains recognizable at both granularities, while the finer configuration subdivides several broad groups into coherent spatial and kinematic subpatterns. The analyst therefore selects a level of granularity appropriate to the intended application, while the membership and observed characteristics of the resulting groups are derived from the trajectory data; domain knowledge is applied subsequently to support their operational interpretation.

The numerical cluster identifiers themselves do not imply equivalence between configurations; for example, Cluster 0 at $K = 5$ and Cluster 0 at $K = 9$ are labels generated independently by the respective clustering solutions. The relationship between the coarse and fine categories is instead established from shared trajectory membership and the continuity of their spatial and kinematic patterns.

The granularity analysis therefore demonstrates an adjustable level of behavioural resolution. A lower value of K is suitable when the analytical objective is to summarize the dominant operational families in a compact form, whereas a higher value can reveal finer within-family distinctions when greater behavioural detail is required. The $K = 5$ configuration is retained for the principal interpretation because it provides a clear high-level categorization, while $K = 9$ demonstrates that the same workflow can reveal coherent operational substructure without changing the underlying trajectory representation. This result satisfies the third success criterion for Experiment #3.

7.6.3 Relationship with conventional aircraft categories

The independent aircraft-reference comparison provides complementary evidence to the behaviour-based categories because the conventional aircraft categories are introduced only after the trajectory groups have been generated. It therefore provides external context for assessing what information the trajectory-derived categories contain relative to conventional classifications based on aircraft configuration and airworthiness information.

The clearest evidence comes from clusters 0, 1, and 3. Medium fixed-wing aircraft account for 93.4% of Cluster 0, 88.7% of Cluster 1, and 92.1% of Cluster 3. Yet Section 7.6.1 identifies these groups as three distinct operational behaviours: runway 13 departures, runway 22 arrivals, and runway 31 departures followed by a turn. Aircraft in the same broad conventional category can therefore fall into different behaviour-based categories depending on how they operate within the observed airspace.

The two higher-altitude groups provide additional context. Cluster 2 contains a substantially larger proportion of large fixed-wing aircraft than the principal terminal groups, whereas Cluster 4 has the most heterogeneous conventional composition. These results indicate that conventional aircraft characteristics and observed trajectory behaviour are related, but they describe different dimensions of the airspace data.

The two categorization approaches are therefore complementary. Conventional airworthiness-related categories primarily describe what an aircraft is (i.e., its physical configuration, size or weight class, propulsion and engine configuration, etc.), whereas behaviour-based categories describe how aircraft operated within the selected airspace during the observation period. Because the aircraft-reference information is introduced after the KMeans assignments have been generated, the initial grouping remains based on trajectory structure, while the independent information supports subsequent interpretation and comparison. This result satisfies the fourth success criterion for Experiment #3.

7.6.4 Response to Research Question #3 and Hypothesis #3

Research Question #3 asks:

How can trajectory-level information from a reusable 4D airspace model be used to identify recurring operationally meaningful patterns in a data-driven manner and support their interpretation when the relevant groupings are not defined in advance?

Experiment #3 answers this question through a workflow in which trajectory-level outputs from Airspace Model #1 are transformed into a common fixed-length representation, grouped via unsupervised clustering, and then interpreted using complementary spatial, kinematic, granularity, and independent aircraft-

reference evidence. Behavioural category labels are therefore assigned after the trajectory groups have been generated, rather than being specified as clustering targets. The data-driven aspect of the approach refers specifically to the generation of cluster membership from the selected trajectory representation; the representation, clustering implementation, and category granularity remain application-specific methodological choices.

Hypothesis #3 states:

Unsupervised clustering applied to trajectory-level outputs from a reusable 4D airspace model can identify recurring aircraft operational patterns directly from observed trajectory structure without predefined behavioural labels, and the resulting data-derived groups can subsequently be interpreted as operationally meaningful behaviour categories using post-clustering spatial, kinematic, and domain evidence.

The combined results support Hypothesis #3 for the evaluated Experiment #3 case study. First, the $K = 5$ KMeans configuration identifies recurring trajectory groups without predefined behavioural category labels. Second, the groups exhibit distinct spatial and kinematic characteristics that support their interpretation as terminal departure, arrival, departure-and-turn, and higher-altitude movement behaviours. Third, the comparison of $K = 5$ and $K = 9$ shows that increasing category granularity reveals coherent, finer subpatterns within the broader trajectory families. Fourth, the independent aircraft-reference comparison shows that operationally distinct behaviour groups can share the same dominant conventional aircraft category, demonstrating that the trajectory-derived categories capture information about observed operation that is complementary to conventional technical classification.

The evidence supporting Hypothesis #3 can therefore be summarized against the four Experiment #3 success criteria established in Section 3.3. Table 25 consolidates the corresponding results and shows how the clustering, spatial and kinematic characterization, granularity analysis, and independent aircraft-reference comparison collectively support the proposed behaviour-based categorization approach.

Table 25: Validation of Experiment #3 results against the success criteria

Success criterion	**Evidence from Experiment #3**
Recurring trajectory groups are identified without predefined behavioural category labels	The 346 trajectory units are assigned to $K = 5$ clusters using the fixed-length trajectory representation before operational behaviour labels or aircraft-reference categories are introduced.
Groups exhibit distinct spatial and kinematic characteristics supporting operational interpretation	Figures 91–97 distinguish the groups using trajectory geometry, altitude evolution, heading, vertical rate, velocity, cluster distribution, and standardized trajectory-level descriptors.
Increased granularity reveals coherent finer operational subpatterns	Figures 98–101 show that broader $K = 5$ groups are subdivided at $K = 9$, including $K = 5$ Cluster 0 into $K = 9$ clusters 0/8 and $K = 5$ Cluster 1 into $K = 9$ clusters 1/6, with corresponding altitude and vertical-rate differences.
Independent aircraft-reference information supports post hoc interpretation and comparison	Figures 102–103 show that operationally distinct clusters 0, 1, and 3 share medium fixed-wing aircraft as their dominant conventional category, demonstrating complementary information between technical classification and observed behaviour.

Taken together, the evidence summarized in Table 25 shows that all four success criteria are met for the evaluated Experiment #3 case study. Experiment #3 therefore demonstrates a second downstream analytical use of the reusable airspace-model outputs established in Experiment #1. Experiment #2 transformed model contents into location-based analytical metrics, whereas Experiment #3 uses trajectory-level outputs for unsupervised operational-pattern discovery. The supporting comparison in Appendix C establishes KMeans as the selected implementation for this application, and the principal Experiment #3 results demonstrate how the resulting data-driven trajectory groups can be transformed into interpretable behaviour-based aircraft categories.

Chapter 8: Discussion

Chapter 8 synthesizes the results of the three research experiments and evaluates how the combined evidence addresses the main research objective stated in Section 1.5: to develop a methodology for a data-driven airspace-modelling approach that leverages big aerospace data using collection and processing techniques effective at managing large-scale datasets while optimizing computational resources. The research addresses this objective through four linked methodological components: i) a modular and reusable DGGS-based 4D airspace representation; ii) computational strategies for generating and reusing that representation while preserving the information required by downstream analyses; iii) a method for deriving multiple location-based analytical products from the reusable representation, demonstrated through multi-hazard risk analysis; and iv) a data-driven operational-pattern identification and interpretation workflow, demonstrated through unsupervised clustering and behaviour-based aircraft categorization. These components were developed and evaluated through the three research experiments.

The thesis addresses the four research gaps summarized in Figure 104. Research Gap #1 concerns the specialization of existing airspace models by aircraft or data type, geographical scale, dimensional representation, or analytical application, which limits their modular reuse across different analyses. Research Gap #2 concerns the computational efficiency of location-based, data-driven 4D airspace models when information required for downstream analyses must be preserved. Research Gap #3 concerns the close coupling between many location-based airspace representations and the analytical purpose for which they were developed, limiting reuse of a common representation for multiple derived analytical products. Research Gap #4 concerns methods for identifying and interpreting recurring operational patterns directly from airspace model data when the relevant groupings are unknown or not defined in advance.

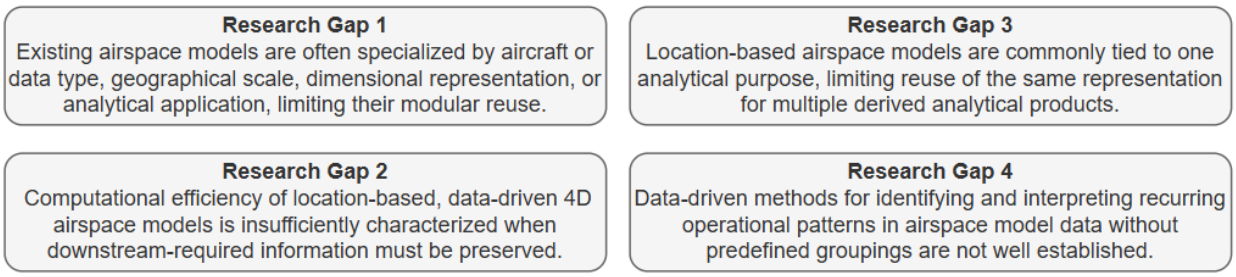


Figure 104: Research gaps addressed by the thesis

The three experiments address these gaps through a common methodological progression. Experiment #1 develops the reusable DGGS-based 4D airspace model and evaluates its computational practicality while preserving the information required for subsequent analyses. Experiment #2 evaluates a downstream use of the model by deriving and interpreting multiple co-indexed analytical products in a location-based multi-hazard risk case study. Experiment #3 evaluates a fundamentally different downstream use by deriving data-driven trajectory groups and subsequently interpreting their operational meaning in a behaviour-based aircraft categorization case study. Together, the experiments progress from constructing and computationally evaluating the common airspace representation to two distinct analytical applications of its outputs.

8.1 Response to the research questions and hypotheses

Figure 105 summarizes how the thesis objective, research gaps, research questions, hypotheses, and experiments are connected. It is included here to provide a thesis-level view of the validation structure before discussing each experiment individually in Sections 8.1.1 to 8.1.3 below.

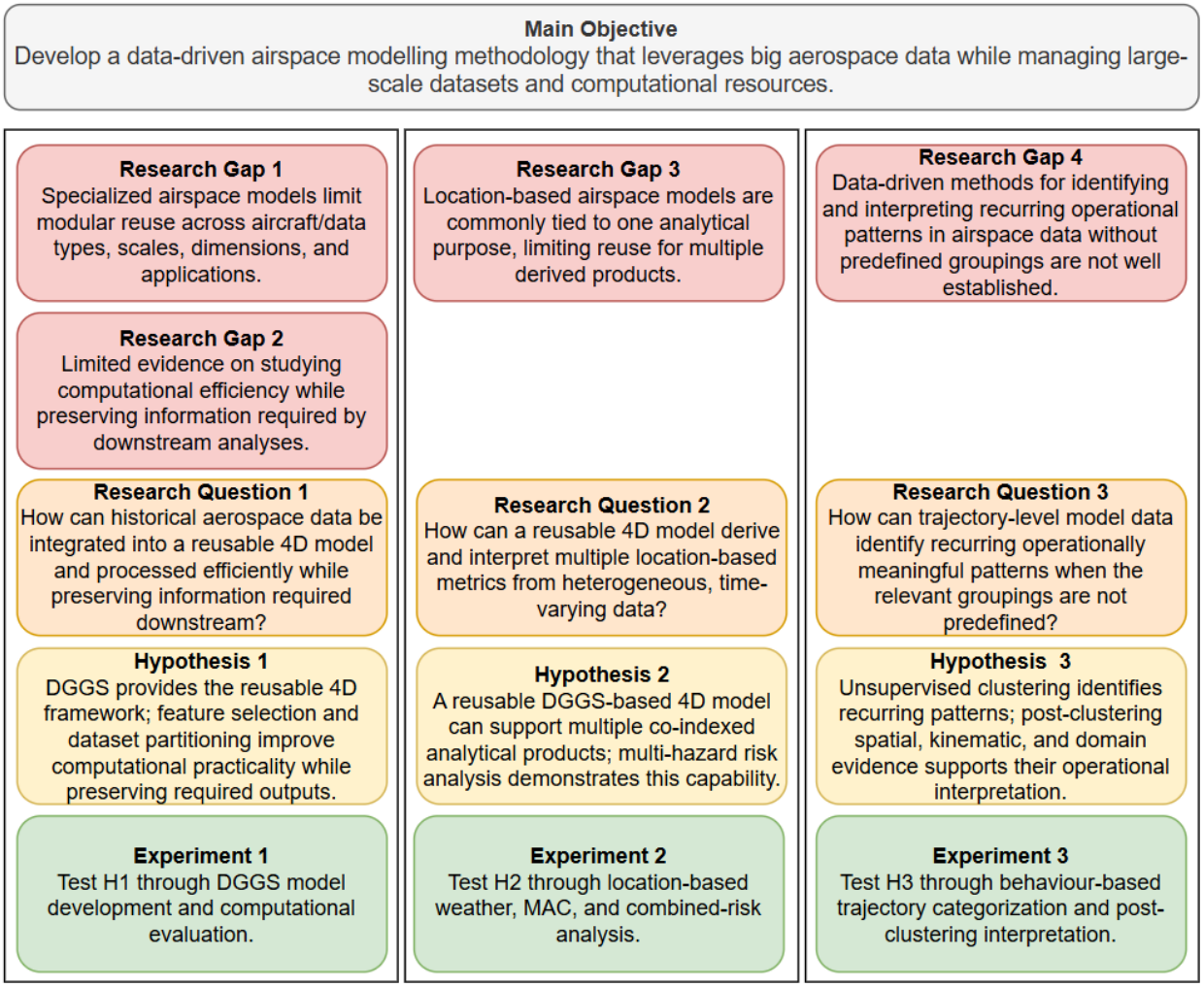


Figure 105: Overview of the validation structure of the thesis

8.1.1 Research Question #1 and Hypothesis #1

Research Gaps #1 and #2 identify two linked limitations. Research Gap #1 concerns the specialization of existing airspace models by aircraft or data type, geographical scale, dimensional representation, or analytical application, which limits their modular reuse across analyses. Research Gap #2 concerns the computational efficiency of location-based, data-driven 4D airspace models when the information required by downstream analyses must be preserved. These gaps led to Research Question #1, which asks how historical aerospace data can be integrated into a generalized, reusable, data-driven 4D airspace model and processed efficiently while preserving the information required for downstream model reuse and analysis. Hypothesis #1 proposed a two-part solution: a DGGS framework for the reusable 4D representation, and

feature selection and dataset partitioning to improve computational practicality while preserving the required model outputs.

Experiment #1 tested this proposal by integrating historical ADS-B aircraft trajectories and NOAA NEXRAD weather data into two DGGS-based airspace-model configurations with different spatiotemporal extents and resolution strategies. The resulting models preserved the spatial, vertical, temporal, and DGGS-indexed information required by the methodology, and two- and three-dimensional visualizations provided evidence that these characteristics remained spatially and operationally coherent. The computational evaluation then compared feature selection, dataset partitioning, and BUILD/LOAD execution behaviour using metrics including runtime, memory demand, stored model-output size, and reuse performance. Output-preservation checks were used to verify that the information required by downstream analyses was retained.

The computational results support a resource-dependent operating strategy. When sufficient memory is available, S2—feature selection without dataset partitioning—is preferred because it reduces BUILD time and substantially reduces the size of saved model outputs without introducing partitioning overhead. Under tighter memory constraints, S4—feature selection with dataset partitioning—provides the preferred alternative because it substantially reduces sustained and peak memory demand at the cost of additional BUILD time. Once reusable model outputs have been generated, LOAD is the most efficient strategy for repeated downstream analyses when model regeneration is unnecessary.

Research Question #1 is therefore answered by demonstrating a reusable DGGS-based method for integrating heterogeneous historical aerospace data and by showing how the resulting model can be generated and reused under different computational constraints while preserving the information required for downstream analyses. Hypothesis #1 is supported for the evaluated datasets and implementation: the DGGS framework provided a common, reusable 4D representation, while feature selection and dataset partitioning affected computational practicality through complementary runtime, storage, and memory mechanisms without materially altering the model outputs required for subsequent analyses.

8.1.2 Research Question #2 and Hypothesis #2

Research Gap #3 concerns the close coupling between many existing airspace representations and the single analytical purpose for which they were developed. This gap led to Research Question #2, which asks how a reusable 4D airspace model can be used to derive and interpret multiple types of location-based airspace metrics from heterogeneous, time-varying airspace data contained within it. Hypothesis #2 proposed that a reusable DGGS-based 4D airspace model can support multiple co-indexed location-based analytical products. A layered risk-analysis case study using aircraft-proximity risk, hazardous-weather risk, and their combined effects was selected to demonstrate this capability.

Using Airspace model #1, Experiment #2 produced three co-indexed analytical products: a weather risk metric derived from storm cell motion, a location-based MAC risk metric derived from aircraft convergence, and a combined weather–MAC risk indicator derived from the two normalized component indicators. The resulting fields varied with the underlying aircraft and weather conditions and were interpreted across location, altitude, and time using time-series, three-dimensional, and two-dimensional visualizations.

Research Question #2 is therefore answered by demonstrating a methodology in which heterogeneous, time-varying airspace data, retained within a common 4D representation, are transformed through application-specific calculations into separate and combined location-based analytical products. Hypothesis #2 is supported for the evaluated risk case study because the aircraft- and weather-derived risk indicators

were independently generated, interpreted, and combined at common DGGS locations and times while retaining the same underlying airspace representation. The experiment therefore provides controlled evidence that a reusable 4D airspace representation can support the derivation, interpretation, and combination of multiple location-based analytical products without redesigning the underlying model.

8.1.3 Research Question #3 and Hypothesis #3

Research Gap #4 concerns methods for identifying and interpreting recurring operational patterns directly from airspace-model data when the relevant groupings are unknown or not defined in advance. Existing approaches often require categories, labels, rules, or analytical boundaries to be established through subject-matter expertise or application-specific assumptions before the underlying structure of the data has been examined. This gap led to Research Question #3: how can trajectory-level information from a reusable 4D airspace model be used to identify recurring, operationally meaningful patterns in a data-driven manner and support their interpretation when the relevant groupings are not defined in advance? Hypothesis #3 proposed that unsupervised clustering can identify recurring aircraft operational patterns from the observed trajectory structure without predefined behavioural labels, and that the resulting data-derived groups can subsequently be interpreted as operationally meaningful behaviour categories using post-clustering spatial, kinematic, and domain evidence.

Experiment #3 evaluated this hypothesis using 346 trajectory units extracted from Airspace Model #1. The trajectories were transformed into a common fixed-length representation and grouped using the selected KMeans implementation. A supporting comparison of DBSCAN, HDBSCAN, KMeans, and GMM, presented in Appendix C and [3], established the clustering implementation for the case study. The principal Experiment #3 validation then examined whether the resulting trajectory groups could be interpreted using spatial and kinematic evidence, how the identified structure changed with category granularity, and how the resulting behaviour-based categories related to independent aircraft-reference information.

The $K = 5$ result identified recurring terminal-area and higher-altitude trajectory groups without predefined behavioural category labels. Post-clustering examination of trajectory geometry, altitude, heading, vertical rate, and velocity supported interpreting these groups as distinct operational behaviours, including runway-aligned departure and arrival families, a departure-and-turn behaviour, and higher-altitude movement patterns. The operational labels were therefore assigned after clustering rather than imposed as clustering targets.

The category-granularity analysis provides an important clarification regarding analyst input. The selected value of K determines the level of behavioural resolution represented by the clustering solution, but it does not prescribe the operational identity or membership of the resulting groups. The $K = 5$ and $K = 9$ results demonstrate this distinction: the principal trajectory structure remains recognizable at both tested granularities, while the finer configuration subdivides several broader groups into coherent spatial and kinematic subpatterns. The $K = 5$ configuration therefore serves as a useful high-level categorization for the selected application rather than a predefined five-category solution.

The independent aircraft-reference comparison provides complementary evidence because conventional aircraft information is introduced only after the trajectory groups have been generated. Several operationally distinct behaviour groups share medium fixed-wing aircraft as their dominant conventional category, demonstrating that the trajectory-derived categories describe how aircraft operated within the selected airspace rather than simply reproducing conventional classifications of what the aircraft are.

Research Question #3 is therefore answered through a workflow in which recurring trajectory structure is first identified from the observed data and operational meaning is subsequently assigned using

complementary domain evidence. Hypothesis #3 is supported for the evaluated case study because the selected unsupervised clustering implementation produced recurring trajectory groups without predefined behavioural labels, and those data-derived groups could subsequently be interpreted using spatial, kinematic, granularity, and independent aircraft-reference evidence. The data-driven character of the approach applies specifically to the generation of group membership from the selected trajectory representation; the representation, clustering implementation, category granularity, and subsequent operational interpretation remain application-specific methodological choices.

8.2 Cross-experiment synthesis

Considered together, the three experiments reveal four complementary findings that are not fully apparent from the conclusions of the individual research questions alone.

First, the three experiments show that model reusability depends on separating the persistent airspace representation from the application-specific analytical methods applied to it. Experiment #1 established a common DGGS-based spatial and temporal representation; Experiment #2 reused that representation to derive and combine aircraft- and weather-related analytical products; and Experiment #3 used trajectory-level outputs from the same modelling workflow for a fundamentally different analytical purpose. Across the experiments, the DGGS structure therefore serves as a stable interface between heterogeneous source observations and downstream analyses, reducing coupling between the underlying airspace representation and any one analytical application.

Second, the experiments show that computational efficiency is meaningful only when the information required by downstream analyses is preserved. Experiment #1 demonstrated that feature selection and dataset partitioning can reduce different computational burdens, while Experiments #2 and #3 demonstrate why preserving the retained information matters: the downstream analyses depend on retaining the relevant aircraft and weather information, along with its spatial and temporal context. A computational improvement should therefore be judged by both resource reduction and the preservation of the data content and spatiotemporal relationships required to answer the intended analytical question.

Third, a reusable airspace representation does not imply a universally fixed analytical configuration. The same modelling framework supports the risk and clustering applications, yet each requires its own analytical formulation, parameter choices, validation criteria, and interpretation procedure. Experiment #1 similarly demonstrates that spatial extent and DGGS resolution should be selected according to analytical requirements rather than treated as universally optimal model properties. Experiment #3 provides a particularly clear example of this distinction. The analyst-selected value of K determines the granularity at which recurring trajectory structure is represented, while the membership and observed characteristics of the resulting groups remain derived from the trajectory data. The coherent operational structure observed at both $K = 5$ and $K = 9$ demonstrates that analyst input can set the desired analytical resolution without predefining the behavioural categories themselves. More generally, the methodology is data-driven rather than assumption-free: application-specific design choices define how the data are represented and analysed, while the observed data determine the resulting analytical structure within those choices. Reusability therefore applies to the common modelling framework, whereas its configuration and downstream analyses remain application-dependent.

Fourth, visualization tools were found to be integral to the evidence-generation process rather than merely a presentation step. Across the three experiments, two- and three-dimensional visualizations were required to verify DGGS construction and mapped source data, relate derived risk indicators to the aircraft and weather conditions from which they were calculated, and determine whether clustering outputs represented

coherent operational behaviours. In several cases, scalar metrics or tabular summaries alone could not establish spatial coherence or operational meaning. Visualization therefore plays an essential cross-cutting role in the verification, diagnosis, interpretation, and communication of complex 4D airspace-model outputs.

8.3 Scope and limitations of the research

The conclusions of the thesis should be interpreted within the scope of the datasets, implementations, and experimental applications used for validation.

8.3.1 Scope of the research

The research is scoped as a methodological evaluation of reusable, data-driven 4D airspace modelling using historical aerospace data. The three experiments evaluate the methodology using selected New York-area airspace environments, publicly accessible ADS-B aircraft trajectories and NEXRAD-derived weather information, and retrospective computational analyses. These choices provide a common experimental basis for evaluating model construction and reuse, location-based analytical-product derivation, and data-driven operational pattern identification.

Within this scope, the risk analysis in Experiment #2 serves as a case study for evaluating the derivation, interpretation, and combination of multiple location-based analytical products from the reusable airspace representation. Experiment #3 similarly uses behaviour-based aircraft categorization as a case study for evaluating data-driven identification of recurring trajectory structure and its subsequent operational interpretation. The computational evaluation is conducted using the implemented Python workflow and selected hardware/software environment. These boundaries define the validation context of the thesis rather than universal operating conditions for every possible implementation or application.

Table 26 summarizes the principal scope boundaries within which the methodology was evaluated. These boundaries define the experimental context of the thesis and are distinct from the limitations discussed in Section 8.3.2.

Table 26: Summary of the experimental scope of the thesis

Scope dimension	Scope adopted in this thesis	Role in the validation
Geographical context	Selected terminal and regional airspace environments in the New York metropolitan area, with detailed analyses centred on LGA	Provides representative dense terminal traffic, multiple spatial scales, and relevant weather interactions
Source data	Historical ADS-B aircraft trajectories and NOAA NEXRAD-derived weather information	Provides heterogeneous, time-varying aerospace data for model construction and downstream analysis
Analysis mode	Retrospective analysis of historical data	Supports controlled and reproducible methodological evaluation
Computational environment	Implemented Python workflow evaluated using the selected hardware and software environment	Provides a common basis for comparing model-generation and reuse strategies
Experiment #2 application	Location-based MAC risk, weather risk, and combined risk analysis	Case study for evaluating derivation and interpretation of multiple co-indexed analytical products
Experiment #3 application	Behaviour-based trajectory categorization using unsupervised clustering and post-clustering interpretation	Case study for evaluating data-driven identification and interpretation of recurring operational patterns

8.3.2 Limitations of the research

Within the scope defined in Section 8.3.1, several limitations constrain the transferability, generalizability, or interpretation of the experimental results. These limitations concern the extent to which conclusions established for the selected datasets, implementations, and case studies can be expected to remain unchanged under varying conditions.

The first limitation concerns geographical transferability. Experiment #1 demonstrates the modelling workflow across two different spatial extents and resolution strategies within the New York metropolitan airspace, while the detailed analytical case studies use selected LGA-centred environments. These results establish that the methodology can operate under different modelling configurations within the evaluated region. However, the transferability of application-specific parameter choices and observed analytical patterns to substantially different airports, en-route environments, traffic structures, weather regimes, and operational procedures requires additional regional validation.

The second limitation concerns dataset transferability. The methodology is evaluated using historical ADS-B aircraft trajectories and NEXRAD-derived weather data. These sources provide the heterogeneous spatial, temporal, aircraft, and weather information required for the three experiments, but the behaviour of the modelling and analytical workflow may differ when other data sources or operational information are introduced. Evaluation with additional sources, such as flight plans, controller intent, airspace restrictions, aircraft-performance information, terrain, or other environmental datasets, would provide further evidence regarding the adaptability of the common 4D representation and its downstream analyses.

The third limitation concerns computational implementation. Experiment #1 evaluates the relative effects of feature selection, dataset partitioning, and BUILD/LOAD execution within the implemented Python workflow and the selected hardware and software environment. The resulting comparisons provide evidence on the computational behaviour of these strategies in a controlled implementation, but absolute

runtime, memory demand, and relative performance may differ under alternative databases, programming implementations, distributed-computing environments, cloud infrastructure, GPU acceleration, or optimized compiled code. The computational conclusions should therefore be interpreted primarily in terms of the relative behaviour of the tested strategies rather than as universal performance values.

The fourth limitation concerns the application-specific formulation of the Experiment #2 risk case study. The normalized weather- and MAC risk indicators use selected convergence-time thresholds that determine their scaling and temporal sensitivity. The combined risk formulation also uses a complementary-product rule that treats the normalized weather and MAC components as independent contributions for combination. These modelling choices provide a transparent basis for evaluating the location-based analytical mechanism, but the resulting values are normalized analytical indicators rather than empirically calibrated event probabilities. Alternative threshold values, dependence structures, and additional aircraft- or weather-related variables would require application-specific calibration and validation.

The fifth limitation concerns the transferability of the behaviour-based categorization results from Experiment #3. The identified categories depend on the selected airspace region, observation period, trajectory representation, preprocessing procedure, clustering implementation, and category granularity. The $K = 5$ and $K = 9$ analyses demonstrate coherent operational structure at two different levels of behavioural resolution, but the appropriate granularity remains dependent on the intended analytical application. The evaluated dataset is also dominated by medium fixed-wing aircraft, while rotorcraft, small fixed-wing aircraft, and some higher-altitude operations are represented by fewer observations. Consequently, the experimental evidence is stronger for recurring behaviours associated with the dominant traffic populations than for less frequently observed vehicles or operations. Evaluation across additional regions, traffic compositions, observation periods, and category granularities would provide stronger evidence regarding the stability and transferability of the identified operational patterns.

The sixth limitation concerns operational deployment. The thesis demonstrates the methodology through retrospective analysis of historical aerospace data and shows that the proposed approach can produce reusable and interpretable outputs within this context. Operational deployment would additionally require validation of real-time operation, integration with operational air traffic management systems, and use within controller, pilot, or planner decision-making. These aspects therefore represent subsequent validation stages before the methodology can be assessed within an operational decision-support environment.

Table 27 summarizes these limitations, the aspect of the conclusions that each constrains, and the principal additional validation required to address it.

Table 27: Summary of the principal limitations of the research and corresponding validation needs

Limitation	Principal constraint on the conclusions	Additional validation
Geographical transferability	Application-specific parameters and observed patterns may differ across airspace environments	Repeat representative analyses in other airports, regions, and traffic/weather conditions
Dataset transferability	Results are established for the selected ADS-B and NEXRAD-based data combination	Evaluate additional heterogeneous operational and environmental data sources
Computational implementation	Absolute performance depends on software and hardware implementation	Replicate computational evaluation using alternative computing architectures and implementations
Risk formulation	Indicator scaling and combination depend on application-specific thresholds and modelling assumptions	Calibrate alternative thresholds, dependence models, and additional hazard variables for target applications
Behaviour-category transferability	Identified patterns and appropriate granularity depend on dataset, region, representation, and traffic composition	Evaluate additional regions, periods, traffic populations, and category granularities
Operational deployment	Retrospective results do not establish real-time performance or operational decision utility	Evaluate near-real-time execution and human/system integration in operational settings

Chapter 9: Conclusion

Chapter 9 concludes the thesis by consolidating the principal conclusions arising from the completed research and identifying directions for future work. The research addressed the main objective stated in Section 1.5: to develop a methodology for a data-driven airspace modelling approach that leverages big aerospace data using collection and processing techniques effective at managing large-scale datasets while optimizing computational resources.

Chapter 8 provided the detailed assessment of the research questions and hypotheses, the cross-experiment synthesis, and the scope and limitations of the experimental validation. This chapter therefore focuses on the principal thesis-level conclusions that emerge from the research, followed by opportunities for extending and further validating the developed methodology.

9.1 Main conclusions

The completed research supports four principal conclusions.

First, a location-based DGGS framework can provide a modular, reusable, four-dimensional representation of heterogeneous historical airspace data. By associating aircraft, weather, and other time-varying observations with a common set of indexed three-dimensional locations and time references, the underlying airspace representation can be decoupled from the application-specific analytical methods subsequently applied to it. The reuse of the resulting model outputs across Experiments #2 and #3 demonstrates the practical value of this separation between a persistent representation and downstream analysis.

Second, computational efficiency in data-driven airspace modelling should be evaluated alongside the preservation of information required by downstream analyses. Experiment #1 showed that feature selection, dataset partitioning, and model reuse address different computational burdens rather than providing a single universally preferable execution strategy. Feature selection reduced processing and storage requirements, dataset partitioning reduced memory demand when needed, and loading previously generated model outputs provided the most efficient approach for repeated downstream analysis. These results demonstrate that computational improvements are meaningful when they reduce resource requirements while retaining the data content and spatiotemporal relationships required by the intended analytical application.

Third, a reusable 4D airspace representation can serve as an analytical foundation from which various location-based products are subsequently derived without redesigning the underlying model. Experiment #2 demonstrated this capability by independently deriving weather risk and aircraft proximity risk indicators and combining them at common DGGS locations and times. The significance of this result is the clear separation between the reusable airspace representation and the application-specific calculations used to generate new analytical information from its contents.

Fourth, trajectory-level outputs from the reusable airspace model can support data-driven identification and subsequent interpretation of recurring operational patterns when the relevant groupings are not defined in advance. Experiment #3 demonstrated this through unsupervised clustering, followed by post-clustering spatial, kinematic, granularity, and independent aircraft-reference analyses. The comparison of $K = 5$ and $K = 9$ further showed that the selected cluster count controls the level of behavioural resolution rather than prescribing the operational identity of the resulting groups. Domain knowledge can therefore be applied after the data-driven grouping process to interpret the resulting structure without defining behavioural categories beforehand.

Across these conclusions, visualization provided a cross-cutting means of verifying model construction, interpreting location-based analytical products, and evaluating the spatial and operational coherence of data-driven patterns. Taken together, the research demonstrates achievement of the main thesis objective through a methodology that combines a reusable 4D airspace representation, computationally practical model generation and reuse, location-based analytical-product derivation, and data-driven operational-pattern identification and interpretation.

9.2 Future work

The scope and limitations identified in Section 8.3 provide several directions for extending and further validating the methodology. Future research can build on the present work through broader external validation, integration of additional airspace data, increased computational scale, extension of downstream analytical applications, further evaluation of behaviour-based operational pattern analysis, and eventual operational implementation.

1. Validate the methodology across additional airspace environments

The methodology could be evaluated across additional terminal, en route, and mixed airspace environments with varying traffic densities, airport geometries, weather regimes, and operational procedures. Applying the same modelling and analytical workflow to geographically distinct datasets would provide stronger evidence on the transferability of DGGS configuration choices, computational behaviour, derived analytical products, and identified operational patterns. Longer observation periods would also allow evaluation of how the identified operational patterns, analytical results, and model characteristics vary under different traffic and weather conditions over time.

2. Expand the airspace entities and data sources represented by the model

Future implementations could extend the common 4D representation to additional historical and operational data, including flight plans, aircraft performance data, airspace restrictions, terrain and obstacles, population exposure, emissions, wind and turbulence fields, UAS operations, and AAM operations. These extensions would provide further evidence that the location-based representation can accommodate heterogeneous entities with diverse spatial, temporal, and physical characteristics while maintaining a common reference structure for downstream analysis.

3. Evaluate computational scalability at larger data and implementation scales

The computational framework could be evaluated using larger geographic domains, longer historical periods, higher data volumes, and alternative computing architectures. Database-backed workflows, parallel or distributed processing, cloud infrastructure, optimized spatial indexing, and compiled or accelerated implementations offer potential avenues for extending the current computational evaluation. These studies should continue to assess efficiency alongside the preservation of information required by downstream analyses, rather than treating runtime or memory reduction alone as the optimization objective.

4. Extend location-based analytical products

The location-based analytical branch of the methodology could be extended by developing additional derived products and further refining the Experiment #2 risk formulation. Future risk applications could evaluate alternative convergence thresholds, incorporate dependencies among contributing hazards, include additional aircraft or meteorological variables, and compare the resulting indicators with established collision risk, conflict detection, or weather-analysis methods where appropriate. More broadly, the same

location-based computation mechanism could be applied to other airspace quantities, such as traffic density, congestion, environmental exposure, occupancy, or operational-complexity measures.

5. Extend validation and application of behaviour-based operational-pattern analysis

Future work could evaluate the behaviour-based categorization approach across different airspace regions, traffic compositions, observation periods, and category granularities. This would provide stronger evidence on the stability and transferability of recurring operational patterns and on the level of granularity appropriate to different analytical objectives. The same general approach could also be extended beyond aircraft trajectories by applying unsupervised learning to other model-derived information, such as location-based risk, traffic density, or environmental fields, to identify recurring spatiotemporal airspace conditions that are difficult to recognize from individual maps or time-series representations alone.

6. Evaluate interactive and operational implementations

The visualization and analytical tools could be extended into an interactive environment that supports three-dimensional exploration, linked spatial and temporal views, dynamic filtering, and interrogation of large model outputs. A subsequent stage of research could evaluate the methodology under near-real-time data ingestion and operational timing requirements, and investigate its integration with air traffic management, planning, safety analysis, or future airspace applications. Such studies would require application-specific validation with operational stakeholders and comparison with the workflows and information requirements of the intended users.

9.3 Final remarks

The broader significance of this research follows from the continuing evolution of the airspace environment described in Chapter 1. Air traffic volumes continue to increase, the quantity and variety of aerospace data available for modelling are expanding, and new aircraft and operational concepts—including UAS, eVTOL vehicles, and Advanced Air Mobility operations—are being introduced alongside conventional aviation. These developments are making airspace environments increasingly heterogeneous, dynamic, data-intensive, and difficult to represent using models designed around a single vehicle type, data source, spatial scale, or analytical application.

Addressing this complexity requires modelling approaches that can integrate heterogeneous information within reusable four-dimensional representations, manage growing computational demands, support different analytical applications from a common underlying model, identify recurring operational structure from data, and provide interpretable representations of complex results. The methodology developed in this thesis demonstrates one implementation of these capabilities using historical airspace data and establishes a foundation for their continued development as airspace operations and the information available to study them evolve.

References

[1] N. Vincent-Boulay and C. Marsden, “Aircraft Categorization Approach Using Machine Learning to Analyze Aircraft Behavior,” *J. Air Transp.*, vol. 32, no. 4, pp. 218–229, 2024, doi: 10.2514/1.D0398.

[2] N. Vincent-Boulay and C. Marsden, “A Geospatial Approach to Modeling Airspace Risk Factors,” *J. Open Aviat. Sci.*, vol. 3, no. 1, Art. no. 1, Feb. 2025, doi: 10.59490/joas.2025.7402.

[3] N. Vincent-Boulay and C. Marsden, “A comparative evaluation methodology of unsupervised clustering algorithms for ADS-B trajectory-based aircraft behavior analysis,” *Comput. Artif. Intell.*, vol. 4, no. 3, 2026.

[4] N. Vincent-Boulay and C. Marsden, “A Novel Approach to Aircraft Categorization using Machine Learning to Analyze Aircraft Behaviour,” presented at the AIAA Aviation 2023, San Diego, 2023. doi: 10.2514/6.2023-4214.

[5] N. Vincent-Boulay and C. Marsden, “Spatial modeling of midair collision risk using ADS-B data,” in *11th OpenSky Symposium*, Toulouse, France, 2023.

[6] N. Vincent-Boulay, C. Marsden, and A. Cui, “Data-driven Geospatial Modeling for Complex Airspace Environments,” presented at the AIAA SCITECH 2024 Forum, 2024, p. 0571. doi: 10.2514/6.2024-0571.

[7] S. S. Chauhan and H. Sharan, “A Review on big data analytics for IOT: Applications and future”.

[8] K. W. K. Soon, C. A. Lee, and P. Boursier, “A chronology of big data adoption: Review of literature,” *IJABER*, vol. 14, no. 1, pp. 521–544, 2016.

[9] D. Laney, “3-D data management: Controlling data volume, velocity and variety. Application Delivery Strategies by META Group Inc,” *Dispon. Sul Sito Available Httpsblogs Gart. Comdoug-Laneyfiles201201ad949-3D-Data-Manag.-Control.-Data-Vol.-Veloc.--Var. Pdf Accessed 6 Febr. 2001*, 2001.

[10] Y. Demchenko, C. Ngo, C. de Laat, P. Membrey, and D. Gordijenko, “Big security for big data: Addressing security challenges for the big data infrastructure,” presented at the Secure Data Management: 10th VLDB Workshop, SDM 2013, Trento, Italy, August 30, 2013, Proceedings 10, Springer, 2014, pp. 76–94.

[11] N. Solutions, “Nalashaa Solutions,” Nalashaa Healthcare Tech Blogs. Accessed: Jan. 27, 2025. [Online]. Available: https://blog.nalashaahealth.com/applications-big-data-healthcare/

[12] N. Abdullah, “The Industrial Internet of Things (IIoT) and its roles in the Fourth Industrial Revolution: A review,” *Indones. J. Comput. Sci.*, vol. 13, no. 2, 2024.

[13] J. Roper, *The Rise of E-Commerce: From Dot to Dominance*. Pen and Sword History, 2023.

[14] “Domo Resource - Data Never Sleeps 5.0.” Accessed: Apr. 17, 2024. [Online]. Available: https://www.domo.com/learn/infographic/data-never-sleeps-5

[15] W. Raghupathi and V. Raghupathi, “Big data analytics in healthcare: promise and potential,” *Health Inf. Sci. Syst.*, vol. 2, no. 1, p. 3, Feb. 2014, doi: 10.1186/2047-2501-2-3.

[16] M. Chen, S. Mao, and Y. Liu, “Big Data: A Survey,” *Mob. Netw. Appl.*, vol. 19, no. 2, pp. 171–209, Apr. 2014, doi: 10.1007/s11036-013-0489-0.

[17] K. Sravanthi and T. S. Reddy, “Applications of Big data in Various Fields,” 2015. Accessed: May 19, 2025. [Online]. Available: https://www.semanticscholar.org/paper/Applications-of-Big-data-in-Various-Fields-Sravanthi-Reddy/987f6f54d3a68a976d76c374476bc648a87f1152?utm_source=consensus

[18] R. Kolya, Prathima, Reemal, A. Raj, M. S. Big, and D. Analytics, “Major Roles of Big Data in Various Fields,” 2020. Accessed: May 19, 2025. [Online]. Available: https://www.semanticscholar.org/paper/MAJOR-ROLES-OF-BIG-DATA-IN-VARIOUS-FIELDS-Kolya-Prathima/541253f463961a86e18af1a4f8f39d97e653fd65?utm_source=consensus

[19] S. S. Roy, P. Samui, R. Deo, and S. Ntalampiras, *Big data in engineering applications*. Springer, 2018.

[20] L. G. Pereira *et al.*, “USE OF BIG DATA IN EDUCATION: HOW ANALYTICS SUPPORTS PEDAGOGICAL DECISIONS,” *LUMEN VIRTUS*, vol. 15, no. 43, Art. no. 43, Nov. 2024, doi: 10.56238/levv15n43-029.
[21] S. D. Dhakal, “Impact of Algorithms and Big Data on Educational Field,” *OCEM J. Manag. Technol. Soc. Sci.*, vol. 4, no. 1, Art. no. 1, Jan. 2025, doi: 10.3126/ocemjmtss.v4i1.74754.
[22] H. Bai, “Design and Application of Decision Support System for Educational Management Based on Big Data,” *J. Electr. Syst.*, vol. 20, no. 6s, Art. no. 6s, Apr. 2024, doi: 10.52783/jes.3084.
[23] V. N. Rao and T. Rao, “Big Data and Its Applications,” *Int. J. Adv. Res. Sci. Commun. Technol.*, 2021, doi: 10.48175/IJARSCT-V2-I3-326.
[24] M. Chen, S. Mao, Y. Zhang, and V. C. M. Leung, “Big Data Applications,” in *Big Data: Related Technologies, Challenges and Future Prospects*, M. Chen, S. Mao, Y. Zhang, and V. C. M. Leung, Eds., Cham: Springer International Publishing, 2014, pp. 59–79. doi: 10.1007/978-3-319-06245-7_6.
[25] G. Sabarmathi and R. Chinnaiyan, “Investigations on big data features research challenges and applications,” in *2017 International Conference on Intelligent Computing and Control Systems (ICICCS)*, Jun. 2017, pp. 782–786. doi: 10.1109/ICCONS.2017.8250569.
[26] A. Seema, “Literature Review on Big Data Applications,” *J. Emerg. Technol. Innov. Res.*, Mar. 2018, Accessed: May 19, 2025. [Online]. Available: https://www.semanticscholar.org/paper/Literature-Review-on-Big-Data-Applications-Seema/0c73948d9adc20a1747ac9e678cdf03ec7253c1b?utm_source=consensus
[27] “Document Search - Web of Science Core Collection.” Accessed: Jun. 16, 2025. [Online]. Available: https://www.webofscience.com/wos/woscc/basic-search
[28] E. F. Adamopoulou and E. Daskalakis, “Applications and Technologies of Big Data in the Aerospace Domain,” *Electronics*, 2023, doi: 10.3390/electronics12102225.
[29] V. E. Badea, A. Zamfiroiu, and R. Boncea, “Big Data in the Aerospace Industry,” vol. 22, pp. 17–24, 2018, doi: 10.12948/ISSN14531305/22.1.2018.02.
[30] G. Burmester, H. Ma, D. Steinmetz, and S. Hartmannn, “Big Data and Data Analytics in Aviation,” in *Advances in Aeronautical Informatics: Technologies Towards Flight 4.0*, U. Durak, J. Becker, S. Hartmann, and N. S. Voros, Eds., Cham: Springer International Publishing, 2018, pp. 55–65. doi: 10.1007/978-3-319-75058-3_5.
[31] A. Corallo, F. O. Buccoliero, A. M. Crespino, V. Del Vecchio, M. Lezzi, and A. Spennato, “Modelling a Big Data-based Analytical Process: an Aerospace Case Study,” in *2023 8th International Conference on Smart and Sustainable Technologies (SpliTech)*, Jun. 2023, pp. 1–6. doi: 10.23919/SpliTech58164.2023.10193675.
[32] S. Ayhan, J. Pesce, P. Comitz, D. Sweet, S. Bliesner, and G. Gerberick, “Predictive analytics with aviation big data,” in *2013 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, Apr. 2013, pp. 1–13. doi: 10.1109/ICNSurv.2013.6548556.
[33] M. Z. Li and M. S. Ryerson, “Reviewing the DATAS of aviation research data: Diversity, availability, tractability, applicability, and sources,” *J. Air Transp. Manag.*, vol. 75, pp. 111–130, 2019.
[34] R. Chennu and V. R. Veeredhi, “Optimization of Aerospace Big Data Including Integrated Health Monitoring With the Help of Data Analytics,” *Res. Anthol. Reliab. Saf. Aviat. Syst. Spacecr. Air Transp.*, 2021, doi: 10.4018/978-1-5225-7077-6.CH004.
[35] V. Bhatia, A. Kumar, S. Sidharth, N. Dogra, F. Altarazi, and J. Singh, “Industry 4.0 in Aircraft Production, Maintenance, Repair, and Overhaul [MRO]: Investigating Use Cases and Associated Patent Landscape,” *Recent Pat. Mech. Eng.*, 2024, doi: 10.2174/0122127976321914240823092031.
[36] K. J. B. Lootens and M. Efthymiou, “The Adoption of Network-Centric Data Sharing in Air Traffic Management,” *Inf. Resour. Manag. J. IRMJ*, vol. 32, no. 3, pp. 48–69, 2019.
[37] R. E. Bach and R. A. Paielli, “A user guide for smoothing air traffic radar data,” National Aeronautics and Space Administration, Ames Research Center, Moffett Field, California, 2014.
[38] Q. Li, Y. Dong, D. Wang, J. Wu, and L. Zhang, “Real-Time Precise DGNSS/INS Integrated Relative Positioning with High Output Rate and Low Broadcast Rate for Kinematic-to-Kinematic Applications,” *Remote Sens*, vol. 14, 2022, doi: 10.3390/rs14092053.

[39] K.-L. Ong *et al.*, "Big Data Applications in Engineering and Science," in *Big Data Concepts, Theories, and Applications*, S. Yu and S. Guo, Eds., Cham: Springer International Publishing, 2016, pp. 315–351. doi: 10.1007/978-3-319-27763-9_9.
[40] B. Singh, S. Kumar, G. Kaur, and M. Kaur, "A Survey on Big Data: Challenges, Tools and Technique.," *Int. J. Adv. Res. Comput. Sci.*, vol. 7, no. 6, 2016.
[41] C. S. Sindhu and N. P. Hegde, "TAF: Temporal Analysis Framework for Handling Data velocity in Healthcare Analytics," in *Proceedings of the International Conference on Big Data and Advanced Wireless Technologies*, in BDAW '16. New York, NY, USA: Association for Computing Machinery, Nov. 2016, pp. 1–11. doi: 10.1145/3010089.3010095.
[42] S. Bagga and A. Sharma, "Big Data and Its Challenges: A Review," in *2018 4th International Conference on Computing Sciences (ICCS)*, Aug. 2018, pp. 183–187. doi: 10.1109/ICCS.2018.00037.
[43] V. Arora, "Data Management Solutions for Tackling Big Data Variety," UC Santa Barbara, 2018. Accessed: May 19, 2025. [Online]. Available: https://escholarship.org/uc/item/15v5r90t
[44] B. Novikov, N. Vassilieva, and A. Yarygina, "Querying big data," in *Proceedings of the 13th International Conference on Computer Systems and Technologies*, in CompSysTech '12. New York, NY, USA: Association for Computing Machinery, Jun. 2012, pp. 1–10. doi: 10.1145/2383276.2383278.
[45] Y. Wu, Y. Xiang, J. Ge, and P. Muller, "High-Performance Computing for Big Data Processing," *Future Gener. Comput. Syst.*, vol. 88, pp. 693–695, Nov. 2018, doi: 10.1016/j.future.2018.07.054.
[46] H. Shu, "Big data analytics: six techniques," *Geo-Spat. Inf. Sci.*, vol. 19, no. 2, pp. 119–128, Apr. 2016, doi: 10.1080/10095020.2016.1182307.
[47] D. Eshmuradov and T. D. Elmuradov, "Mathematical modelling of aeronautical environment," *Civ. Aviat. High Technol.*, 2020, doi: 10.26467/2079-0619-2020-23-5-67-75.
[48] I. Roychoudhury *et al.*, "Real-time monitoring and prediction of airspace safety," 2018.
[49] D. Guimarans, P. Arias, M. Tomasella, and C.-L. Wu, "Chapter 4 - A Review of Sustainability in Aviation: A Multidimensional Perspective," in *Sustainable Transportation and Smart Logistics*, J. Faulin, S. E. Grasman, A. A. Juan, and P. Hirsch, Eds., Elsevier, 2019, pp. 91–121. doi: 10.1016/B978-0-12-814242-4.00004-1.
[50] H. Ming-hua, "Probabilistic airspace congestion management model and methodology," *J. Shandong Univ.*, 2010, Accessed: Dec. 04, 2024. [Online]. Available: https://consensus.app/papers/probabilistic-airspace-congestion-management-model-and-ming-hua/540b82809f335f1f9c4044e44996e3e4/
[51] S. Ayhan and H. Samet, "Aircraft Trajectory Prediction Made Easy with Predictive Analytics," in *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*, in KDD '16. New York, NY, USA: Association for Computing Machinery, Aug. 2016, pp. 21–30. doi: 10.1145/2939672.2939694.
[52] V. P. Jilkov, J. H. Ledet, and X. R. Li, "Multiple Model Method for Aircraft Conflict Detection and Resolution in Intent and Weather Uncertainty," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 55, no. 2, pp. 1004–1020, Apr. 2019, doi: 10.1109/TAES.2018.2867698.
[53] A. Alharbi, I. Petrunin, and D. Panagiotakopoulos, "Traffic Flow Prediction for UTM Application: A Deep Learning Approach," *2022 IEEEAIAA 41st Digit. Avion. Syst. Conf. DASC*, pp. 1–10, 2022, doi: 10.1109/DASC55683.2022.9925774.
[54] C. Wanke, S. Zobell, and L. Song, "Probabilistic Airspace Congestion Management," 2005, doi: 10.2514/6.2005-7433.
[55] C. Bongiorno, S. Micciché, and R. N. Mantegna, "An empirically grounded agent based model for modeling directs, conflict detection and resolution operations in air traffic management," *PLOS ONE*, vol. 12, no. 4, p. e0175036, Apr. 2017, doi: 10.1371/journal.pone.0175036.
[56] S. B. Amor and M. Bui, "A COMPLEX SYSTEM APPROACH IN MODELING AIRSPACE CONGESTION DYNAMICS," *J. Air Transp. Stud.*, vol. 3, no. 1, Art. no. 1, Jan. 2012, doi: 10.38008/jats.v3i1.95.

[57] F. Ho *et al.*, "Pre-Flight Conflict Detection and Resolution for UAV Integration in Shared Airspace: Sendai 2030 Model Case," *IEEE Access*, vol. 7, pp. 170226–170237, 2019, doi: 10.1109/ACCESS.2019.2954987.
[58] Z. Liu, K. Cai, J. Xie, and Y. Zhu, "A Network-Based Conflict Resolution Approach for Unmanned Aerial Vehicle Operations in Dense Nonsegregated Airspace," *IEEE Intell. Transp. Syst. Mag.*, vol. 14, pp. 212–232, 2022, doi: 10.1109/MITS.2021.3049420.
[59] M. Schäfer, M. Strohmeier, V. Lenders, I. Martinovic, and M. Wilhelm, "Bringing up OpenSky: A large-scale ADS-B sensor network for research," presented at the IPSN-14 Proceedings of the 13th International Symposium on Information Processing in Sensor Networks, IEEE, 2014, pp. 83–94. doi: 10.1109/IPSN.2014.6846743.
[60] J. Sun, J. M. Hoekstra, and J. Ellerbroek, "OpenAP: An open-source aircraft performance model for air transportation studies and simulations," *Aerospace*, vol. 7, no. 8, p. 104, 2020.
[61] A. Weinert, N. Underhill, B. Gill, and A. Wicks, "Processing of Crowdsourced Observations of Aircraft in a High Performance Computing Environment," presented at the 2020 IEEE High Performance Extreme Computing Conference (HPEC), IEEE, 2020, pp. 1–6.
[62] A. Weinert, N. Underhill, C. Serres, and R. Guendel, "Correlated Bayesian Model of Aircraft Encounters in the Terminal Area Given a Straight Takeoff or Landing," *Aerospace*, vol. 9, no. 2, p. 58, 2022, doi: https://doi.org/10.3390/aerospace9020058.
[63] R. Alligier, "Predictive Distribution of Mass and Speed Profile to Improve Aircraft Climb Prediction," *J. Air Transp.*, vol. 28, no. 3, pp. 114–123, 2020, doi: 10.2514/1.D0181.
[64] J. Sun, J. Ellerbroek, and J. M. Hoekstra, "Reconstructing aircraft turn manoeuvres for trajectory analyses using ads-b data," *Proc. 9th SESAR Innov. Days*, pp. 2–5, 2019.
[65] T. Bolić, L. Castelli, A. De Lorenzo, and F. Vascotto, "Trajectory Clustering for Air Traffic Categorisation," *Aerospace*, vol. 9, no. 5, Art. no. 5, May 2022, doi: 10.3390/aerospace9050227.
[66] Federal Aviation Administration, "FAA Aerospace Forecast Fiscal Years 2021–2041," 2021. [Online]. Available: https://www.faa.gov/data_research/aviation/aerospace_forecasts/
[67] M. Gelhausen, P. Berster, and D. Wilken, "Modelling future airport capacity enlargements and limits," pp. 189–204, 2020, doi: 10.1016/b978-0-12-812657-8.00008-7.
[68] I. Galarraga, L. M. Abadie, T. Standfuss, I. R. de Gauna, and N. Goicoechea, "Estimating the Volatility of Flights and Risk of Saturation of Airspaces in the European Core Area: A Methodological Proposal," *Appl. Sci.*, 2023, doi: 10.3390/app132312576.
[69] J. Rosenow and H. Fricke, "Impact of multi-criteria optimized trajectories on European airline efficiency, safety and airspace demand," *J. Air Transp. Manag.*, 2019, doi: 10.1016/J.JAIRTRAMAN.2019.01.001.
[70] S. Ayhan, Í. R. De Oliveira, G. Balvedi, P. Costas, A. Leite, and F. C. F. De Azevedo, "Big data-driven prediction of airspace congestion," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*, Oct. 2023, pp. 1–8. doi: 10.1109/DASC58513.2023.10311134.
[71] W. Torres-Pomales, "Toward a Safety Risk-Based Classification of Unmanned Aircraft," NASA Langley Research Center, TM–2016-219367, 2016.
[72] W. B. Cotton, "Adaptive Autonomous Separation for Uam in Mixed Operations," in *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, Apr. 2019, pp. 1–11. doi: 10.1109/ICNSURV.2019.8735196.
[73] B. G. Franciscone and E. Fernandes, "Challenges to the Operational Safety and Security of eVTOL Aircraft in Metropolitan Regions: A Literature Review," *J. Airl. Oper. Aviat. Manag.*, 2023, doi: 10.56801/jaoam.v2i1.2.
[74] D. Sengupta and S. K. Das, "Urban Air Mobility: Vision, Challenges and Opportunities," *2023 IEEE 24th Int. Conf. High Perform. Switch. Routing HPSR*, pp. 1–6, 2023, doi: 10.1109/HPSR57248.2023.10148014.
[75] X. Yang, L. Deng, J. Liu, P. Wei, and H. Li, "Multi-Agent Autonomous Operations in Urban Air Mobility with Communication Constraints," 2020, doi: 10.2514/6.2020-1839.

[76] N. Pongsakornsathien, S. Bijjahalli, A. Gardi, A. Symons, Y. Xi, and R. Sabatini, "A Performance-Based Airspace Model for Unmanned Aircraft Systems Traffic Management," *Aerospace*, vol. 7, no. 11, Art. no. 11, Nov. 2020, doi: 10.3390/aerospace7110154.
[77] "Current Landscape of Unmanned Aircraft Systems at Airports," Transportation Research Board, Washington, D.C., Dec. 2019. doi: 10.17226/25659.
[78] The National Academies of Sciences, "Assessing the risks of integrating unmanned aircraft systems into the national airspace system," The National Academies Press, Washington, DC, 2018. doi: 10.17226/25143.
[79] E. National Academies of Sciences, *In-Time Aviation Safety Management: Challenges and Research for an Evolving Aviation System*. 2018. doi: 10.17226/24962.
[80] L. Xi, Z. Jun, Z. Yanbo, and L. Wei, "Simulation study of algorithms for aircraft trajectory prediction based on ADS-B technology," in *2008 Asia Simulation Conference - 7th International Conference on System Simulation and Scientific Computing*, Oct. 2008, pp. 322–327. doi: 10.1109/ASC-ICSC.2008.4675378.
[81] W. Yang, J. Tang, R. He, and Y. Chen, "A Medium-Term Conflict Detection and Resolution Method for Open Low-Altitude City Airspace Based on Temporally and Spatially Integrated Strategies," *IEEE Trans. Control Syst. Technol.*, vol. 28, no. 5, pp. 1817–1830, Sep. 2020, doi: 10.1109/TCST.2019.2925579.
[82] W. Liu and I. Hwang, "Probabilistic 4D trajectory prediction and conflict detection for air traffic control," in *49th IEEE Conference on Decision and Control (CDC)*, Dec. 2010, pp. 1183–1188. doi: 10.1109/CDC.2010.5717681.
[83] Z. Wang, H. Li, J. Wang, and F. Shen, "Deep reinforcement learning based conflict detection and resolution in air traffic control," *IET Intell. Transp. Syst.*, 2019, doi: 10.1049/IET-ITS.2018.5357.
[84] S. Ayhan, P. Costas, and H. Samet, "A Data-driven Framework for Long-Range Aircraft Conflict Detection and Resolution," *ACM Trans. Spat. Algorithms Syst. TSAS*, vol. 5, pp. 1–23, 2019, doi: 10.1145/3328832.
[85] F. Ho, R. Geraldes, A. Gonçalves, M. Cavazza, and H. Prendinger, "Improved Conflict Detection and Resolution for Service UAVs in Shared Airspace," *IEEE Trans. Veh. Technol.*, vol. 68, pp. 1231–1242, 2019, doi: 10.1109/TVT.2018.2889459.
[86] S. Hao, S. Cheng, and Y. Zhang, "A multi-aircraft conflict detection and resolution method for 4-dimensional trajectory-based operation," *Chin. J. Aeronaut.*, 2018, doi: 10.1016/J.CJA.2018.04.017.
[87] Y. I. Jenie, E.-J. van Kampen, J. Ellerbroek, and J. M. Hoekstra, "Safety Assessment of a UAV CD&R System in High Density Airspace Using Monte Carlo Simulations," *IEEE Trans. Intell. Transp. Syst.*, vol. 19, no. 8, pp. 2686–2695, Aug. 2018, doi: 10.1109/TITS.2017.2758859.
[88] Y. Chen and D. Zhang, "Dynamic airspace configuration method based on a weighted graph model," *Chin. J. Aeronaut.*, vol. 27, pp. 903–912, 2014, doi: 10.1016/J.CJA.2014.06.009.
[89] J. Kim and E. Atkins, "Airspace Geofencing and Flight Planning for Low-Altitude, Urban, Small Unmanned Aircraft Systems," *Appl. Sci.*, 2022, doi: 10.3390/app12020576.
[90] G. Zhu and P. Wei, "Low-Altitude UAS Traffic Coordination with Dynamic Geofencing," in *16th AIAA Aviation Technology, Integration, and Operations Conference*, in AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics, Jun. 2016. doi: 10.2514/6.2016-3453.
[91] C. H. J. Wang, S. K. Tan, and K. Low, "Three-dimensional (3D) Monte-Carlo modeling for UAS collision risk management in restricted airport airspace," *Aerosp. Sci. Technol.*, vol. 105, 2020, doi: 10.1016/j.ast.2020.105964.
[92] A. Pritchett, S. Lee, D. Huang, and D. Goldsman, "Hybrid-System Simulation for National Airspace System Safety Analysis," presented at the Journal of Aircraft, Aug. 2000, pp. 1132–1142. doi: 10.1109/WSC.2000.899076.
[93] J. Xiang, J. Chen, and Y. Liu, "Hybrid Multiscale Search for Dynamic Planning of Multi-Agent Drone Traffic," *J. Guid. Control Dyn.*, vol. 46, no. 10, pp. 1963–1974, Oct. 2023, doi: 10.2514/1.G007343.

[94] N. Musavi, A. Manzoor, and Y. Yildiz, “A 3D Game Theoretical Framework for the Evaluation of Unmanned Aircraft Systems Airspace Integration Concepts,” *ArXiv*, vol. abs/1802.07218, 2018, doi: 10.1016/j.trc.2021.103417.
[95] C.-E. Lin and Y. Lai, “UAV Path Prediction for CD&R to Manned Aircraft in a Confined Airspace for Cooperative Mission,” *Int. J. Aerosp. Eng.*, vol. 2018, 2018, doi: 10.1155/2018/8759836.
[96] N. Zhang, M. Zhang, and K. Low, “3D path planning and real-time collision resolution of multirotor drone operations in complex urban low-altitude airspace,” *Transp. Res. Part C-Emerg. Technol.*, vol. 129, 2021, doi: 10.1016/J.TRC.2021.103123.
[97] B. Xia, I. Mantegh, and W. Xie, “UAV Multi-Dynamic Target Interception: A Hybrid Intelligent Method Using Deep Reinforcement Learning and Fuzzy Logic,” *Drones*, 2024, doi: 10.3390/drones8060226.
[98] R. Shone, K. Glazebrook, and K. G. Zografos, “Applications of stochastic modeling in air traffic management: Methods, challenges and opportunities for solving air traffic problems under uncertainty,” *Eur. J. Oper. Res.*, vol. 292, no. 1, pp. 1–26, Jul. 2021, doi: 10.1016/j.ejor.2020.10.039.
[99] Z. Zhao, J. Yuan, and Y. Liu, “Research and comparison on identification and prediction methods of air traffic network congestion,” in *2022 4th International Academic Exchange Conference on Science and Technology Innovation (IAECST)*, Dec. 2022, pp. 1256–1264. doi: 10.1109/IAECST57965.2022.10061994.
[100] L. Adacher, M. Flamini, and E. Romano, “Rerouting algorithms solving the air traffic congestion,” *AIP Conf. Proc.*, vol. 1836, no. 1, p. 020053, Jun. 2017, doi: 10.1063/1.4981993.
[101] X. Dai, M. Hu, W. Tian, and H. Liu, “Modeling Congestion Propagation in Multistage Schedule within an Airport Network,” *J. Adv. Transp.*, vol. 2018, no. 1, p. 6814348, 2018, doi: 10.1155/2018/6814348.
[102] F. N. Sutton and K. S. Lindsay, “Workload-based congestion prediction,” in *2015 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, Apr. 2015, pp. S4-1-S4-9. doi: 10.1109/ICNSURV.2015.7121261.
[103] L. Yang, “Stochastic Traffic Flow Modeling and Optimal Congestion Pricing.,” Thesis, 2012. Accessed: Jun. 16, 2025. [Online]. Available: http://deepblue.lib.umich.edu/handle/2027.42/94054
[104] H. Lee and H. Balakrishnan, “A comparison of two optimization approaches for airport taxiway and runway scheduling,” in *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, Oct. 2012, pp. 4E1-1-4E1-15. doi: 10.1109/DASC.2012.6382348.
[105] S. Shresta *et al.*, “UAS in The NAS for Simulation Modeling,” in *2021 Integrated Communications Navigation and Surveillance Conference (ICNS)*, Apr. 2021, pp. 1–8. doi: 10.1109/ICNS52807.2021.9441590.
[106] A. R. Odoni and R. W. Simpson, “Review and evaluation of national airspace system models,” Cambridge, MA : Flight Transportation Laboratory, Dept. of Aeronautics and Astronautics, Massachusetts Institute of Technology ; 1979, Technical Report, 1979. Accessed: Jan. 25, 2025. [Online]. Available: https://dspace.mit.edu/handle/1721.1/67941
[107] D. Bertsimas and S. S. Patterson, “The Air Traffic Flow Management Problem with Enroute Capacities,” *Oper. Res.*, vol. 46, no. 3, pp. 406–422, Jun. 1998, doi: 10.1287/opre.46.3.406.
[108] A. Marzuoli, E. Boidot, and E. Feron, “Large-scale data-based collaborative air traffic optimization for congestion management,” in *2013 IEEE/AIAA 32nd Digital Avionics Systems Conference (DASC)*, Oct. 2013, pp. 1C2-1-1C2-16. doi: 10.1109/DASC.2013.6712514.
[109] M. C. R. Murca and M. de Oliveira, “A Data-Driven Probabilistic Trajectory Model for Predicting and Simulating Terminal Airspace Operations,” *2020 AIAAIEEE 39th Digit. Avion. Syst. Conf. DASC*, pp. 1–7, 2020, doi: 10.1109/DASC50938.2020.9256644.
[110] X. Guan, R. Lv, L. Sun, and Y. Liu, “A study of 4D trajectory prediction based on machine deep learning,” in *2016 12th World Congress on Intelligent Control and Automation (WCICA)*, Jun. 2016, pp. 24–27. doi: 10.1109/WCICA.2016.7578458.
[111] G. Hunter, “A probabilistic trajectory constraint modeler,” *2016 IEEEAIAA 35th Digit. Avion. Syst. Conf. DASC*, pp. 1–9, 2016, doi: 10.1109/DASC.2016.7777992.

[112] C. Magaña and E. Juan, “Trajectory prediction uncertainty modelling for Air Traffic Management,” 2016. Accessed: Jun. 16, 2025. [Online]. Available: https://www.semanticscholar.org/paper/Trajectory-prediction-uncertainty-modelling-for-Air-Maga%C3%B1a-Juan/0511975b2c4b4262652d97b7af1706d362313f47?utm_source=consensus

[113] C. E. Verdonk Gallego, V. F. Gómez Comendador, M. A. Amaro Carmona, R. M. Arnaldo Valdés, F. J. Sáez Nieto, and M. García Martínez, “A machine learning approach to air traffic interdependency modelling and its application to trajectory prediction,” *Transp. Res. Part C Emerg. Technol.*, vol. 107, pp. 356–386, Oct. 2019, doi: 10.1016/j.trc.2019.08.015.

[114] Y.-K. Kim, D. G. Lee, and J. W. Han, “Trajectory Calculation Based on Position and Speed for Effective Air Traffic Flow Management,” in *Multimedia and Ubiquitous Engineering*, J. J. (Jong H. Park, J. K.-Y. Ng, H.-Y. Jeong, and B. Waluyo, Eds., Dordrecht: Springer Netherlands, 2013, pp. 377–384. doi: 10.1007/978-94-007-6738-6_47.

[115] Y. K. Kim, J. W. Han, and H. Park, “Trajectory Prediction for Using Real Data and Real Meteorological Data,” in *Ubiquitous Computing Application and Wireless Sensor*, J. J. (Jong H. Park, Y. Pan, H.-C. Chao, and G. Yi, Eds., Dordrecht: Springer Netherlands, 2015, pp. 89–103. doi: 10.1007/978-94-017-9618-7_9.

[116] F. Asadi and A. Richards, “Self-Organized Model Predictive Control for Air Traffic Management,” in *Proceedings of the 5th International Conference on Application and Theory of Automation in Command and Control Systems*, in ATACCS ’15. New York, NY, USA: Association for Computing Machinery, Sep. 2015, pp. 151–159. doi: 10.1145/2899361.2899377.

[117] Y.-J. Chiang, J. T. Klosowski, C. Lee, and J. Mitchell, “Geometric algorithms for conflict detection/resolution in air traffic management,” *Proc. 36th IEEE Conf. Decis. Control*, vol. 2, pp. 1835–18402, 1997, doi: 10.1109/CDC.1997.657848.

[118] J. Murphy, R. Reisman, J. Clayton, and R. Wright, “Physics-Based and Parametric Trajectory Prediction Performance Comparison for Traffic Flow Management,” in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, in Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 2003. doi: 10.2514/6.2003-5629.

[119] B. Sawhill, J. Herriot, B. Holmes, and N. Alexandrov, “Complexity Science Applications to Dynamic Trajectory Management: Research Strategies,” in *9th AIAA Aviation Technology, Integration, and Operations Conference (ATIO)*, in Aviation Technology, Integration, and Operations (ATIO) Conferences. American Institute of Aeronautics and Astronautics, Sep. 2009. doi: 10.2514/6.2009-7090.

[120] H. Chen, V. P. Jilkov, and X. R. Li, “On Threshold Optimization for Aircraft Conflict Detection,” presented at the 18th International Conference on Information Fusion, 2015.

[121] M. Prandini, J. Hu, J. Lygeros, and S. Sastry, “A probabilistic approach to aircraft conflict detection,” *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 199–220, 2000.

[122] D. Cecchi, L. Pollini, and M. Innocenti, “Probabilistic bounds in air traffic conflict resolution,” *Proc. 2003 Am. Control Conf. 2003*, vol. 6, pp. 5388–53936, 2003, doi: 10.1109/ACC.2003.1242585.

[123] A. Lecchini, W. Glover, J. Lygeros, and J. Maciejowski, “Monte Carlo Optimisation for Conflict Resolution in Air Traffic Control,” in *Stochastic Hybrid Systems: Theory and Safety Critical Applications*, H. A. P. Blom and J. Lygeros, Eds., Berlin, Heidelberg: Springer, 2006, pp. 257–276. doi: 10.1007/11587392_8.

[124] J. K. Kuchar and L. C. Yang, “A review of conflict detection and resolution modeling methods,” *IEEE Trans. Intell. Transp. Syst.*, vol. 1, no. 4, pp. 179–189, Dec. 2000, doi: 10.1109/6979.898217.

[125] T. Williamson and N. Spencer, “Development and operation of the Traffic Alert and Collision Avoidance System (TCAS),” *Proc IEEE*, vol. 77, pp. 1735–1744, 1989, doi: 10.1109/5.47735.

[126] M. Chen, “Flight Conflict Detection and Resolution Based on Neural Network,” *2011 Int. Conf. Comput. Inf. Sci.*, pp. 860–862, 2011, doi: 10.1109/ICCIS.2011.154.

[127] H. Wang, Q. Zhang, J. Wu, S. Pan, and Y. Chen, “Time series feature learning with labeled and unlabeled data,” *Pattern Recognit.*, vol. 89, pp. 55–66, May 2019, doi: 10.1016/j.patcog.2018.12.026.

[128] Q. Xu, Z. Chen, F. Li, Z. Shen, and W. Wei, "An Efficient Aircraft Conflict Detection and Resolution Method Based on an Improved Reinforcement Learning Framework," *Int. J. Aerosp. Eng.*, 2023, doi: 10.1155/2023/6643903.
[129] Hojjat Emami and Farnaz Derakhshan, "A New Method for Conflict Detection and Resolution in Air Traffic Management," AAAI Technical Report WS-12-13. Accessed: Jun. 16, 2025. [Online]. Available: https://aaai.org/papers/aaaiw-ws0821-12-5214/
[130] Alfons Geser, César Munoz, Gilles Dowek, and Florent Kirchner, "Air Traffic Conflict Resolution and Recovery," NASA, NASA/CR-2002-211637, 2002.
[131] Q. Xu, J. Huang, Z. Liu, and H. Ding, "A Method Based on Deep Reinforcement Learning to Generate Control Strategy for Aircrafts in Terminal Sector," in *Artificial Intelligence in China*, Q. Liang, W. Wang, J. Mu, X. Liu, Z. Na, and X. Cai, Eds., Singapore: Springer, 2021, pp. 356–363. doi: 10.1007/978-981-15-8599-9_41.
[132] A. Bastas and G. A. Vouros, "Data-Driven Modeling of Air Traffic Controllers' Policy to Resolve Conflicts," *Aerospace*, vol. 10, no. 6, Art. no. 6, Jun. 2023, doi: 10.3390/aerospace10060557.
[133] P. Menon, G. Sweriduk, T. Lam, G. M. Diaz, and K. Bilimoria, "Computer-Aided Eulerian Air Traffic Flow Modeling and Predictive Control," *J. Guid. Control Dyn.*, vol. 29, pp. 12–19, 2004, doi: 10.2514/1.13496.
[134] D. Sun, "Large-scale modeling and optimization of en route air traffic flow," University of California, Berkeley, 2008.
[135] B. Sridhar, S. R. Grabbe, and A. Mukherjee, "Modeling and Optimization in Traffic Flow Management," *Proc. IEEE*, vol. 96, no. 12, pp. 2060–2080, Dec. 2008, doi: 10.1109/JPROC.2008.2006141.
[136] A. Bayen, P. Grieder, G. Meyer, and C. J. Tomlin, "Lagrangian Delay Predictive Model for Sector-Based Air Traffic Flow," *J. Guid. Control Dyn.*, vol. 28, no. 5, pp. 1015–1026, Sep. 2005, doi: 10.2514/1.15242.
[137] P. K. Menon, G. D. Sweriduk, and K. D. Bilimoria, "New Approach for Modeling, Analysis, and Control of Air Traffic Flow," *J. Guid. Control Dyn.*, vol. 27, no. 5, pp. 737–744, Sep. 2004, doi: 10.2514/1.2556.
[138] B. Sridhar and P. K. Menon, "COMPARISON OF LINEAR DYNAMIC MODELS FOR AIR TRAFFIC FLOW MANAGEMENT," *IFAC Proc. Vol.*, vol. 38, no. 1, pp. 13–18, Jan. 2005, doi: 10.3182/20050703-6-CZ-1902.01963.
[139] P. k. Menon, G. Sweriduk, T. Lam, V. Cheng, and K. Bilimoria, "Air Traffic Flow Modeling, Analysis and Control," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, in Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 2003. doi: 10.2514/6.2003-5712.
[140] D. Sun and A. M. Bayen, "Multicommodity Eulerian-Lagrangian Large-Capacity Cell Transmission Model for En Route Traffic," *J. Guid. Control Dyn.*, vol. 31, no. 3, pp. 616–628, May 2008, doi: 10.2514/1.31717.
[141] M. Bielli, G. Calicchio, S. Iozzia, and B. Nicoletti, "Air Traffic Flow Control Systems: Modelling and Evaluation," *IFAC Proc. Vol.*, vol. 14, no. 2, pp. 2357–2362, Aug. 1981, doi: 10.1016/S1474-6670(17)63820-5.
[142] A. P. Saraf and G. L. Slater, "Optimal dynamic scheduling of aircraft arrivals at congested airports," *J. Guid. Control Dyn.*, vol. 31, no. 1, pp. 53–65, 2008.
[143] R. L. Raffard and C. J. Tomlin, "Second order adjoint-based optimization of ordinary and partial differential equations with application to air traffic flow," in *Proceedings of the 2005, American Control Conference, 2005.*, Jun. 2005, pp. 798–803 vol. 2. doi: 10.1109/ACC.2005.1470057.
[144] I. S. Strub and A. M. Bayen, "Continuous adjoint method for Air Traffic Flow Management," in *Proceedings of the 45th IEEE Conference on Decision and Control*, Dec. 2006, pp. 101–106. doi: 10.1109/CDC.2006.377458.

[145] V. Aditya, D. S. Aswin, S. V. Dhaneesh, S. Chakravarthy, B. S. Kumar, and M. Venkadavarahan, "A review on air traffic flow management optimization: trends, challenges, and future directions," *Discov. Sustain.*, vol. 5, no. 1, p. 519, Dec. 2024, doi: 10.1007/s43621-024-00781-7.
[146] B. Sridhar, G. B. Chatterji, and A. D. Evans, "Lessons Learned in the Application of Machine Learning Techniques to Air Traffic Management," in *AIAA AVIATION 2020 FORUM*, American Institute of Aeronautics and Astronautics. doi: 10.2514/6.2020-2882.
[147] A. A. Alharbi, I. Petrunin, and D. Panagiotakopoulos, "Modeling and Characterization of Traffic Flow Patterns and Identification of Airspace Density for UTM Application," *IEEE Access*, vol. 10, pp. 130110–130134, 2022, doi: 10.1109/ACCESS.2022.3228828.
[148] A. M. Bayen, R. L. Raffard, and C. J. Tomlin, "Eulerian network model of air traffic flow in congested areas," in *Proceedings of the 2004 American Control Conference*, Jun. 2004, pp. 5520–5526 vol.6. doi: 10.23919/ACC.2004.1384733.
[149] S. George *et al.*, "Build 8 of the Airspace Concept Evaluation System," in *AIAA Modeling and Simulation Technologies Conference*, in Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 2011. doi: 10.2514/6.2011-6373.
[150] Zhipeng Wei and Jing Li, "Design of Air Traffic Control Operation System Using Multi Agent Technology and Simulation," *Open Fuels Energy Sci. J.*, vol. 7, no. 1, p. 1, 2015, doi: 10.2174/1876973X01508010368.
[151] C. Wang, J. Ge, and X. Xu, "Analysis of Air Traffic Flow Control through Agent-Based Modeling and Simulation," in *2009 International Conference on Computer Modeling and Simulation*, Feb. 2009, pp. 286–290. doi: 10.1109/ICCMS.2009.38.
[152] W. Glover and J. Lygeros, "A Stochastic Hybrid Model for Air Traffic Control Simulation," in *Hybrid Systems: Computation and Control*, R. Alur and G. J. Pappas, Eds., Berlin, Heidelberg: Springer, 2004, pp. 372–386. doi: 10.1007/978-3-540-24743-2_25.
[153] C. E. Seah and I. Hwang, "Stochastic Linear Hybrid Systems: Modeling, Estimation, and Application in Air Traffic Control," *IEEE Trans. Control Syst. Technol.*, vol. 17, no. 3, pp. 563–575, May 2009, doi: 10.1109/TCST.2008.2001377.
[154] M. Kamgarpour, "Optimal Control of Hybrid Systems in Air Traffic Applications," UC Berkeley, 2011. Accessed: Jun. 16, 2025. [Online]. Available: https://escholarship.org/uc/item/4x0827vx
[155] R. Medianto, H. M. Pasaribu, and H. Muhammad, "Development of Hybrid Simulation Model of Air Traffic Management in the Terminal Control Area," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 645, no. 1, p. 012003, Oct. 2019, doi: 10.1088/1757-899X/645/1/012003.
[156] S. McGovern and A. Kalish, "Stochastic airspace simulation tool development," in *2009 IEEE/AIAA 28th Digital Avionics Systems Conference*, 2009, pp. 25–1. doi: 10.1109/DASC.2009.5347537.
[157] J. Krozel and N. Doble, "Simulation of the National Airspace System in Inclement Weather," 2007, doi: 10.2514/6.2007-6558.
[158] S. Roy, Y. Wan, C. Taylor, and C. Wanke, "A Stochastic Network Model for Uncertain Spatiotemporal Weather Impact at the Strategic Time Horizon," in *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, in Aviation Technology, Integration, and Operations (ATIO) Conferences. American Institute of Aeronautics and Astronautics, Sep. 2010. doi: 10.2514/6.2010-9348.
[159] C. Pasaoglu, B. Baspinar, N. K. Ure, and G. Inalhan, "Hybrid systems modeling and automated air traffic control for three-dimensional separation assurance," *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.*, vol. 230, no. 9, pp. 1788–1809, Jul. 2016, doi: 10.1177/0954410015619443.
[160] B. Başpınar, H. Balakrishnan, and E. Koyuncu, "Optimization-Based Autonomous Air Traffic Control for Airspace Capacity Improvement," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 56, no. 6, pp. 4814–4830, Dec. 2020, doi: 10.1109/TAES.2020.3003106.
[161] B. Sridhar, K. Bilimoria, S. Grabbe, G. Chatterji, K. Sheth, and D. Mulfinger, "Software for Simulating Air Traffic," ARC-14653-1, Sep. 2006. Accessed: Jun. 16, 2025. [Online]. Available: https://ntrs.nasa.gov/citations/20110013117

[162] Y. Tian, B. Ye, M. S. Estupiñá, and L. Wan, "Stochastic Simulation Optimization for Route Selection Strategy Based on Flight Delay Cost," *Asia-Pac. J. Oper. Res.*, vol. 35, no. 06, p. 1850045, Dec. 2018, doi: 10.1142/S0217595918500458.
[163] J. Dangermond, *What is a Geographic Information System (GIS)?* ASTM International, 1992.
[164] A. Panayotov, "High resolution airport airspace model using geographic information system," unav, 2009.
[165] L. Mitsevich and N. Zhukovskaya, "Geospatial modeling, analysis and mapping for aerodrome land development," *E3S Web Conf.*, vol. 310, p. 04003, 2021, doi: 10.1051/e3sconf/202131004003.
[166] A. Panayotov, I. Georgiev, and I. Georgiev, "A practical approach for airport spatial modeling," in *Proceedings of the 13th International Conference on Computer Systems and Technologies*, in CompSysTech '12. New York, NY, USA: Association for Computing Machinery, Jun. 2012, pp. 321–328. doi: 10.1145/2383276.2383323.
[167] S.-W. Chang, "A GIS model for analyzing airspace obstructions and safety near airports," *J. Civ. Eng. Archit.*, vol. 10, pp. 553–62, 2016.
[168] M. Mekni, "Integration of GIS Data for Visualization of Virtual Geospatial Environments," *2015 Second Int. Conf. Math. Comput. Sci. Ind. MCSI*, pp. 273–282, 2015, doi: 10.1109/MCSI.2015.39.
[169] T. Bernhardsen, *Geographic information systems: an introduction*. John Wiley & Sons, 2002.
[170] M. F. Goodchild *et al.*, "Next-generation digital earth," *Proc. Natl. Acad. Sci.*, vol. 109, no. 28, pp. 11088–11094, 2012.
[171] M. F. Goodchild, "Discrete global grids for digital earth," presented at the International Conference on Discrete Global Grids, Mar. 2000. Accessed: May 23, 2023. [Online]. Available: https://escholarship.org/uc/item/4tm0d1jm
[172] M. Li and E. Stefanakis, "Geospatial operations of discrete global grid systems—A comparison with traditional GIS," *J. Geovisualization Spat. Anal.*, vol. 4, pp. 1–21, 2020.
[173] H. Djavaherpour, A. Mahdavi-Amiri, and F. F. Samavati, "Physical Visualization of Geospatial Datasets," *IEEE Comput. Graph. Appl.*, vol. 37, no. 3, pp. 61–69, May 2017, doi: 10.1109/MCG.2017.38.
[174] Y. Jie-qing, "Global 3D-Grid Based on Sphere Degenerated Octree and Its Distortion Features," *Geogr. Geo-Inf. Sci.*, 2009, Accessed: Jun. 16, 2025. [Online]. Available: https://www.semanticscholar.org/paper/Global-3D-Grid-Based-on-Sphere-Degenerated-Octree-Jie-qing/b6cb44a91b6b26fdae181efcc64e7e96c3eadc06?utm_source=consensus
[175] C. Deng, C. Cheng, T. Qu, S. Li, and B. Chen, "A Method for Managing ADS-B Data Based on a 4D Airspace-Temporal Grid (GeoSOT-AS)," *Aerospace*, vol. 10, no. 3, p. 217, Mar. 2023, doi: 10.3390/aerospace10030217.
[176] W. Zhai, X. Tong, S. Miao, C. Cheng, and F. Ren, "Collision Detection for UAVs Based on GeoSOT-3D Grids," *ISPRS Int. J. Geo-Inf.*, vol. 8, no. 7, Art. no. 7, Jul. 2019, doi: 10.3390/ijgi8070299.
[177] D. Sahadevan Neelakandan and H. A. Ali, "Enhancing Trajectory-Based Operations for UAVs through Hexagonal Grid Indexing: A Step towards 4D Integration of UTM and ATM," *Int. J. Aviat. Aeronaut. Aerosp.*, vol. 10, no. 2, Jan. 2023, doi: 10.58940/2374-6793.1815.
[178] W. Zhai, B. Han, D. Li, J. Duan, and C. Cheng, "A low-altitude public air route network for UAV management constructed by global subdivision grids," *PLOS ONE*, vol. 16, no. 4, p. e0249680, Apr. 2021, doi: 10.1371/journal.pone.0249680.
[179] Q. Liu, X. Zhao, X. Wang, M. Qin, and W. Sun, "A Multi-Scale Airspace Sectorization Framework Based on QTM and HDQN," *Aerospace*, vol. 12, no. 6, p. 552, Jun. 2025, doi: 10.3390/aerospace12060552.
[180] J. Aryal, R. Raffin, A. Francois, and M. Daniel, "3D modelling and analysis: ISO standard tools for air traffic," vol. 2, pp. 327–334, 2012.
[181] H. Panchal, R. Khan, S. Sengupta, and N. Sarda, "GIS-based Smart Campus System using 3D Modeling," 2011. Accessed: Jun. 16, 2025. [Online]. Available: https://www.semanticscholar.org/paper/GIS-based-Smart-Campus-System-using-3D-Modeling-Panchal-Khan/aeb08916df5c5a898a5b20dcc3462012b2379531?utm_source=consensus

[182] Z. Li, X. Guan, R. Li, and H. Wu, "4D-SAS: A Distributed Dynamic-Data Driven Simulation and Analysis System for Massive Spatial Agent-Based Modeling," *ISPRS Int. J. Geo-Inf.*, vol. 5, no. 4, Art. no. 4, Apr. 2016, doi: 10.3390/ijgi5040042.
[183] T. Chailungka, P. Horma, C. Juthamanee, W. Vongsantivanich, and S. Chaimatanan, "Geo-Information platform for visualisation, analysis and optimization for aviation," in *2017 10th International Conference on Ubi-media Computing and Workshops (Ubi-Media)*, Aug. 2017, pp. 1–4. doi: 10.1109/UMEDIA.2017.8074111.
[184] W. Wright, M. Shields, E. Doucet, and D. W. Rowe, "Airspace Operational Efficiency: Transformation Through GIS-Enabled Applications," in *2024 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, Apr. 2024, pp. 1–10. doi: 10.1109/ICNS60906.2024.10550609.
[185] G. Sun, T. Qu, B. Han, H. Sun, and Q. Zhang, "Research on Airspace Grid Modeling Based on GeoSOT Global Subdivision Model," in *2021 IEEE 3rd International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, Oct. 2021, pp. 501–505. doi: 10.1109/ICCASIT53235.2021.9633363.
[186] C. Conrad, Y. Xu, D. Panda, and A. Tsourdos, "Intelligent Vertiport Traffic Flow Management for Scalable Advanced Air Mobility Operations," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*, Oct. 2023, pp. 1–10. doi: 10.1109/DASC58513.2023.10311299.
[187] K. Namuduri, "Digital Twin Approach for Integrated Airspace Management With Applications to Advanced Air Mobility," *IEEE Open J. Veh. Technol.*, vol. 4, pp. 693–700, 2023, doi: 10.1109/OJVT.2023.3312277.
[188] D. W. Matolak and R. Sun, "Air–Ground Channel Characterization for Unmanned Aircraft Systems—Part III: The Suburban and Near-Urban Environments," *IEEE Trans. Veh. Technol.*, vol. 66, no. 8, pp. 6607–6618, Aug. 2017, doi: 10.1109/TVT.2017.2659651.
[189] J. Cornelius and S. Schmitz, "Rotor Performance Predictions for UAM—Single vs Coaxial Rigid Rotors," presented at the VFS Aeromechanics for Advanced Vertical Flight Technical Meeting, 2022.
[190] K. Song, "Optimal Vertiport Airspace and Approach Control Strategy for Urban Air Mobility (UAM)," *Sustainability*, vol. 15, no. 1, Art. no. 1, Jan. 2023, doi: 10.3390/su15010437.
[191] I. C. Kleinbekman, M. Mitici, and P. Wei, "eVTOL Arrival Sequencing and Scheduling for On-Demand Urban Air Mobility," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, 2018, pp. 1–7. doi: 10.1109/DASC.2018.8569645.
[192] H. Heramarwan, N. Müller, R. Hann, and T. Lutz, "UAM Icing: Ice Accretion Experiments and CFD Icing Simulations on Rotors for eVTOL Unmanned Aircraft," presented at the International Conference on Icing of Aircraft, Engines, and Structures, SAE International, Jun. 2023. doi: 10.4271/2023-01-1391.
[193] M. M. Gomaa, R. Sabatini, and A. Gardi, "AAM and UAS Collision Avoidance in the Presence of Wind and Wake Turbulence," in *2023 IEEE Aerospace Conference*, Mar. 2023, pp. 1–10. doi: 10.1109/AERO55745.2023.10115593.
[194] J.-P. B. Clarke, S. Solak, L. Ren, and A. E. Vela, "Determining Stochastic Airspace Capacity for Air Traffic Flow Management," *Transp. Sci.*, vol. 47, no. 4, pp. 542–559, Nov. 2013, doi: 10.1287/trsc.1120.0440.
[195] Y. Xie, A. Gardi, R. Sabatini, and M. Liang, "Hybrid AI-based Dynamic Re-routing Method for Dense Low-Altitude Air Traffic Operations," in *2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC)*, Sep. 2022, pp. 1–9. doi: 10.1109/DASC55683.2022.9925777.
[196] U. Choi and S. Lee, "Optimization of collision-free trajectories for advanced air mobility under risk of a non-cooperative intruder," in *2023 14th International Conference on Information and Communication Technology Convergence (ICTC)*, Oct. 2023, pp. 378–380. doi: 10.1109/ICTC58733.2023.10393649.
[197] A. Mukherjee, "Dynamic Stochastic Optimization Models for Air Traffic Flow Management," 2004. Accessed: Jun. 17, 2025. [Online]. Available: https://escholarship.org/uc/item/2vk8w6nc

[198] R. Han, H. Li, E. J. Knoblock, M. R. Gasper, and R. D. Apaza, "Joint Velocity and Spectrum Optimization in Urban Air Transportation System via Multi-Agent Deep Reinforcement Learning," *IEEE Trans. Veh. Technol.*, vol. 72, no. 8, pp. 9770–9782, Aug. 2023, doi: 10.1109/TVT.2023.3256067.

[199] L. Haoyang and V. Kharchenko, "Air Traffic Control and Flight Planning for Cargo UAVS in a Single Airspace," *Electron. Control Syst.*, vol. 4, no. 78, Art. no. 78, Dec. 2023, doi: 10.18372/1990-5548.78.18280.

[200] M. C. R. Murça, W. S. S. Souza, and J. V. T. Ribeiro, "Characterizing Performance Tradeoffs from Air Traffic Management Services for Advanced Air Mobility," *J. Aerosp. Inf. Syst.*, vol. 22, no. 2, pp. 116–129, Feb. 2025, doi: 10.2514/1.I011469.

[201] R. Bridgelall, S. White, and D. Tolliver, "Integrating Electric Vertical Takeoff and Landing Aircraft into Public Airspace: A Scenario Study," *Future Transp.*, vol. 3, no. 3, Art. no. 3, Sep. 2023, doi: 10.3390/futuretransp3030057.

[202] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. E. Robinson, "Unmanned Aircraft System Traffic Management (UTM) Concept of Operations," presented at the AIAA AVIATION Forum and Exposition, Washington, DC, Jun. 2016. Accessed: Jun. 17, 2025. [Online]. Available: https://ntrs.nasa.gov/citations/20190000370

[203] J. T. Luxhøj, "A Socio-technical Model for Analyzing Safety Risk of Unmanned Aircraft Systems (UAS): An Application to Precision Agriculture," *Procedia Manuf.*, vol. 3, pp. 928–935, Jan. 2015, doi: 10.1016/j.promfg.2015.07.140.

[204] M. Kochenderfer, L. Espindle, M. Edwards, J. Kuchar, and J. Griffith, "Airspace encounter models for conventional and unconventional aircraft," presented at the Proceedings of 8th USA/Europe Air Traffic Management Research and Development Seminar, USA, 2009.

[205] S. S. Borener, D. Hufty, V. S. Guzhva, K. Martin, and R. Fraga, "Modeling emergent risks in complex airspace: UAS operations in a metroplex environment," in *2015 IEEE/AIAA 34th Digital Avionics Systems Conference (DASC)*, Sep. 2015, pp. 5B3-1-5B3-13. doi: 10.1109/DASC.2015.7311428.

[206] A. Dixit, J. Falusi, S. Kim, G. Savit, and R. Stengel, "Flight Testing a UAV in Simulated Controlled Airspace," in *AIAA Guidance, Navigation, and Control (GNC) Conference*, in Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 2013. doi: 10.2514/6.2013-4701.

[207] Z. Zhao *et al.*, "Mission-Aware Spatio-Temporal Deep Learning Model for UAS Instantaneous Density Prediction," in *2020 International Joint Conference on Neural Networks (IJCNN)*, Jul. 2020, pp. 1–8. doi: 10.1109/IJCNN48605.2020.9207016.

[208] A. Weinert and N. Underhill, "Generating Representative Small UAS Trajectories using Open Source Data," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, Sep. 2018, pp. 1–10. doi: 10.1109/DASC.2018.8569745.

[209] S. Shresta, "UAS traffic and flight plans for simulation modeling," in *2018 Integrated Communications, Navigation, Surveillance Conference (ICNS)*, Apr. 2018, pp. 1–19. doi: 10.1109/ICNSURV.2018.8384932.

[210] P. Pothana, J. Joy, P. Snyder, and S. Vidhyadharan, "UAS Air-Risk Assessment In and Around Airports," in *2023 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, Apr. 2023, pp. 1–11. doi: 10.1109/ICNS58246.2023.10124319.

[211] R. DeLaura, M. Robinson, M. Pawlak, and J. Evans, "Modeling convective weather avoidance in enroute airspace," presented at the 13th Conference on Aviation, Range, and Aerospace Meteorology, AMS, New Orleans, LA, Citeseer, 2008.

[212] M. Steiner and J. Krozel, "Translation of ensemble-based weather forecasts into probabilistic air traffic capacity impact," *2009 IEEEAIAA 28th Digit. Avion. Syst. Conf.*, pp. 26–1, 2009, doi: 10.1109/DASC.2009.5347538.

[213] A. Klein and L. Cook, "Three models for weather impacted airspace capacity estimation and forecast," presented at the Proc. 9th USA/Europe Air Traffic Management R&D Seminar (ATM2011), Berlin, Germany, 2011.

[214] H. Iskenderian *et al.*, "Using Satellite Data to Improve Convective Forecasts in the Consolidated Storm Prediction for Aviation (CoSPA)," in *1st AIAA Atmospheric and Space Environments Conference*, in Fluid Dynamics and Co-located Conferences. American Institute of Aeronautics and Astronautics, Jun. 2009. doi: 10.2514/6.2009-3530.

[215] J. Post, D. Murphy, and J. Bonn, "A Regression Model of National Airspace System Delay," in *6th AIAA Aviation Technology, Integration and Operations Conference (ATIO)*, in Aviation Technology, Integration, and Operations (ATIO) Conferences. American Institute of Aeronautics and Astronautics, Sep. 2006. doi: 10.2514/6.2006-7746.

[216] B. Sridhar and N. Y. Chen, "Short-Term National Airspace System Delay Prediction Using Weather Impacted Traffic Index," *J. Guid. Control Dyn.*, vol. 32, no. 2, pp. 657–662, Mar. 2009, doi: 10.2514/1.38798.

[217] J. Mitchell, V. Polishchuk, and J. Krozel, "Airspace Throughput Analysis Considering Stochastic Weather," in *AIAA Guidance, Navigation, and Control Conference and Exhibit*, in Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 2006. doi: 10.2514/6.2006-6770.

[218] M. V. McCrea, H. D. Sherali, and A. A. Trani, "A probabilistic framework for weather-based rerouting and delay estimations within an Airspace Planning model," *Transp. Res. Part C Emerg. Technol.*, vol. 16, no. 4, pp. 410–431, Aug. 2008, doi: 10.1016/j.trc.2007.09.001.

[219] R. DeLaura, B. Crowe, R. Ferris, J. Love, and W. Chan, "Comparing convective weather avoidance models and aircraft-based data," presented at the 89th Annual Meeting of the American Meteorological Society: Aviation, Range and Aerospace Meteorology Special Symposium on Weather–Air Traffic Impacts, Phoenix, AZ, 2009.

[220] J. Zhou, H. Zhang, W. Lyu, J. Wan, J. Zhang, and W. Song, "Hybrid 4-Dimensional Trajectory Prediction Model, Based on the Reconstruction of Prediction Time Span for Aircraft en Route," *Sustainability*, vol. 14, no. 7, Art. no. 7, Jan. 2022, doi: 10.3390/su14073862.

[221] J. Tang, M. A. Piera, and T. Guasch, "Coloured Petri net-based traffic collision avoidance system encounter model for the analysis of potential induced collisions," *Transp. Res. Part C Emerg. Technol.*, vol. 67, pp. 357–377, Jun. 2016, doi: 10.1016/j.trc.2016.03.001.

[222] A. Alonso-Ayuso, L. F. Escudero, and F. J. Martín-Campo, "A mixed 0–1 nonlinear optimization model and algorithmic approach for the collision avoidance in ATM: Velocity changes through a time horizon," *Comput. Oper. Res.*, vol. 39, no. 12, pp. 3136–3146, Dec. 2012, doi: 10.1016/j.cor.2012.03.015.

[223] J. Mollinga and H. V. Hoof, "An Autonomous Free Airspace En-route Controller using Deep Reinforcement Learning Techniques," *ArXiv*, Jul. 2020, Accessed: May 21, 2025. [Online]. Available: https://www.semanticscholar.org/paper/An-Autonomous-Free-Airspace-En-route-Controller-Mollinga-Hoof/f432dae7e0a9c84adeef9e154e10f8ccab9d5731

[224] C. Weng, C. Chen, J. Tan, T. Pan, and R. Zhong, "Real-time Traffic Simulation and Management for Large-scale Urban Air Mobility: Integrating Route Guidance and Collision Avoidance," Dec. 02, 2024, *arXiv*: arXiv:2412.01235. doi: 10.48550/arXiv.2412.01235.

[225] B. M. Artacho, "Unmanned aircraft system (UAS) integration to airspace and collision risk assessment," masters, Memorial University of Newfoundland, 2018. Accessed: May 21, 2025. [Online]. Available: https://research.library.mun.ca/13082/

[226] K. Jang, Y. V. Pant, A. Rodionova, and R. Mangharam, "Learning-to-Fly RL: Reinforcement Learning-based Collision Avoidance for Scalable Urban Air Mobility," in *2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC)*, Oct. 2020, pp. 1–10. doi: 10.1109/DASC50938.2020.9256710.

[227] Y. Zheng *et al.*, "Design of Fine Management System for Civil Aviation Airspace Resources Based on Spatiotemporal Grid Model," in *2019 IEEE 1st International Conference on Civil Aviation Safety and Information Technology (ICCASIT)*, Oct. 2019, pp. 547–551. doi: 10.1109/ICCASIT48058.2019.8973205.

[228] S. Oh and Y. Yoon, “Data-driven risk analysis of unmanned aircraft system operations considering spatiotemporal characteristics of population distribution,” *Transp. Res. Interdiscip. Perspect.*, vol. 16, p. 100732, Dec. 2022, doi: 10.1016/j.trip.2022.100732.
[229] A. Pilko, A. Sóbester, J. P. Scanlan, and M. Ferraro, “Spatiotemporal Ground Risk Mapping for Uncrewed Aircraft Systems Operations,” *J. Aerosp. Inf. Syst.*, vol. 20, no. 3, pp. 126–139, Mar. 2023, doi: 10.2514/1.I011113.
[230] M. D. Tandale, J. Kwan, S. Lin, and V. V. Vaddi, “Modeling and Simulation Tools for Analysis of Terminal Airspace Operations,” in *2013 Aviation Technology, Integration, and Operations Conference*, in AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics, Aug. 2013. doi: 10.2514/6.2013-4259.
[231] D. Gianazza, C. Allignol, and N. Saporito, “An efficient airspace configuration forecast,” presented at the ATM 2009, 8th USA/Europe Air Traffic Management Research and Developpment Seminar, Jun. 2009. Accessed: May 19, 2025. [Online]. Available: https://enac.hal.science/hal-01020720
[232] M. Hickey and J. Hoffman, “Hybrid Queuing-Regression Model Application for Airspace Analysis,” in *2019 Integrated Communications, Navigation and Surveillance Conference (ICNS)*, Apr. 2019, pp. 1–12. doi: 10.1109/ICNSURV.2019.8735137.
[233] N. Wang *et al.*, “Study of time-dependent queuing models of the national airspace system,” *Comput. Ind. Eng.*, vol. 117, pp. 108–120, Mar. 2018, doi: 10.1016/j.cie.2018.01.014.
[234] B. Han, T. Qu, Z. Huang, Q. Wang, and X. Pan, “Emergency airport site selection using global subdivision grids,” *Big Earth Data*, vol. 6, no. 3, pp. 276–293, Jul. 2022, doi: 10.1080/20964471.2021.1996866.
[235] X. Wang, P. P.-J. Yang, M. Balchanos, and D. Mavris, “Urban Airspace Route Planning for Advanced Air Mobility Operations,” in *Intelligence for Future Cities*, R. Goodspeed, R. Sengupta, M. Kyttä, and C. Pettit, Eds., in The Urban Book Series. Cham: Springer Nature Switzerland, 2023, pp. 193–211. doi: 10.1007/978-3-031-31746-0_11.
[236] E. Pensado *et al.*, “Dynamic Graph Advanced Air Mobility Trajectory Optimization Under Weather Forecast Uncertainty,” *J. Guid. Control Dyn.*, vol. 47, pp. 1–13, Aug. 2024, doi: 10.2514/1.G007971.
[237] J. M. Wunderli, C. Zellmann, M. Köpfli, and M. Habermacher, “sonAIR – a GIS-Integrated Spectral Aircraft Noise Simulation Tool for Single Flight Prediction and Noise Mapping,” *Acta Acust. United Acust.*, vol. 104, no. 3, pp. 440–451, May 2018, doi: 10.3813/AAA.919180.
[238] M. Ha *et al.*, “CMS: A Computational Memory Solution for High-Performance and Power-Efficient Recommendation System,” in *2022 IEEE 4th International Conference on Artificial Intelligence Circuits and Systems (AICAS)*, Jun. 2022, pp. 491–494. doi: 10.1109/AICAS54282.2022.9869851.
[239] V. Calo, S. Badia, and J. Principe, “Editorial: Algorithmic Aspects of High-Performance Computing for Mechanics and Physics,” *J. Comput. Appl. Math.*, vol. 344, p. 739, Dec. 2018, doi: 10.1016/j.cam.2018.07.001.
[240] D. Rajan, O. I. Khalaf, and T. Udayabanu, “Streamlined Algorithms for Large-Scale Matrix Computations in High-Performance Computing”.
[241] H. X. Lin and T. Chan, “Combining the power of high speed computer systems and efficient algorithms,” *Future Gener. Comput. Syst.*, vol. 18, no. 1, p. 1, Sep. 2001, doi: 10.1016/S0167-739X(00)00089-3.
[242] G. Schryen, “Speedup and efficiency of computational parallelization: A unifying approach and asymptotic analysis,” Nov. 15, 2023, *arXiv*: arXiv:2212.11223. doi: 10.48550/arXiv.2212.11223.
[243] R. Manikandan, A. Professor, and D. A. M. Kalpana, “A study on feature selection in big data,” *2017 Int. Conf. Comput. Commun. Inform. ICCCI*, pp. 1–5, 2017, doi: 10.1109/ICCCI.2017.8117751.
[244] J. Li *et al.*, “Feature Selection,” *ACM Comput. Surv. CSUR*, vol. 50, pp. 1–45, 2016, doi: 10.1145/3136625.
[245] Z. Wang, X. Xiao, and S. Rajasekaran, “Novel and efficient randomized algorithms for feature selection,” *Big Data Min Anal*, vol. 3, pp. 208–224, 2020, doi: 10.26599/bdma.2020.9020005.

[246] D. Liu and X. Yue, "Efficient Feature Selection Algorithm Based on Counterfactuals," *2024 6th Int. Conf. Commun. Inf. Syst. Comput. Eng. CISCE*, pp. 541–544, 2024, doi: 10.1109/CISCE62493.2024.10653192.
[247] G. T. Kalayci, A. F. Alkaya, and R. Algin, "Exploitation and Comparison of Computational Intelligence Techniques on the Feature Selection Problem," *Intell. Fuzzy Tech. Big Data Anal. Decis. Mak.*, 2019, doi: 10.1007/978-3-030-23756-1_146.
[248] K. Patidar and D. Tiwari, "Feature Selection using Multi-Objective Clustering based Gray Wolf Optimization for Big Data Analytics," *2022 2nd Int. Conf. Technol. Adv. Comput. Sci. ICTACS*, pp. 122–128, 2022, doi: 10.1109/ICTACS56270.2022.9987754.
[249] N. Alamsyah, V. Restreva, D. Fakultas, I. Akbar, and E. Setiana, "Optimizing Computational Efficiency in Feature Selection for Machine Learning Models: A Study Crime Detection Based on Criminal Data," *2023 Eighth Int. Conf. Inform. Comput. ICIC*, pp. 1–6, 2023, doi: 10.1109/ICIC60109.2023.10382057.
[250] J. Saelee *et al.*, "Enhanced Feature Selection via Hierarchical Concept Modeling," *Appl. Sci.*, 2024, doi: 10.3390/app142310965.
[251] "Chunking as the result of an efficiency computation trade-off," *Nat. Commun.*, vol. 7, 2016, doi: 10.1038/ncomms12176.
[252] C. Yang and J. Chen, "A Scalable Data Chunk Similarity Based Compression Approach for Efficient Big Sensing Data Processing on Cloud," *IEEE Trans. Knowl. Data Eng.*, vol. 29, pp. 1144–1157, 2017, doi: 10.1109/TKDE.2016.2531684.
[253] W. Xia, D. Feng, H. Jiang, Y. Zhang, V. Chang, and X. Zou, "Accelerating content-defined-chunking based data deduplication by exploiting parallelism," *Future Gener Comput Syst*, vol. 98, pp. 406–418, 2019, doi: 10.1016/J.FUTURE.2019.02.008.
[254] S. Misailovic, "Accuracy-aware optimization of approximate programs," *2015 Int. Conf. Compil. Archit. Synth. Embed. Syst. CASES*, pp. 37–38, 2015, doi: 10.1109/CASES.2015.7324543.
[255] H. M. M. Ali, M. Kod, A. M. Al-Salim, and C. Al-Amiri, "Algorithm for Greening Big Data Networks in Iraq Under Veracity Dimension," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 671, 2020, doi: 10.1088/1757-899X/671/1/012052.
[256] D. P. Thipphavong *et al.*, "Initial Study of Tailored Trajectory Management for Multi-Vehicle Uncrewed Regional Air Cargo Operations," in *2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC)*, Oct. 2023, pp. 1–10. doi: 10.1109/DASC58513.2023.10311269.
[257] W. Tian and M. Hu, "The Application of Multi-Objective Generic Algorithm in the Airspace Flow Program," American Society of Civil Engineers, Apr. 2012, pp. 2219–2224. doi: 10.1061/41039(345)367.
[258] C. Cheng *et al.*, "Machine-Learning-Aided Trajectory Prediction and Conflict Detection for Internet of Aerial Vehicles," *IEEE Internet Things J.*, vol. 9, no. 8, pp. 5882–5894, Apr. 2022, doi: 10.1109/JIOT.2021.3060904.
[259] M. Gharibi, "On the Integration of Unmanned Aerial Vehicles into Public Airspace," University of Waterloo, 2020. [Online]. Available: http://hdl.handle.net/10012/15943
[260] A. Fernandes, D. Wesely, B. Holtzman, D. Sweet, and N. Ghazavi, "Using Synchronized Trajectory Data to Improve Airspace Demand Predictions," in *2020 Integrated Communications Navigation and Surveillance Conference (ICNS)*, Sep. 2020, pp. 3C2-1-3C2-15. doi: 10.1109/ICNS50378.2020.9222909.
[261] I. Panchal, S. F. Armanini, and I. C. Metz, "Validation of Collision Detection and Avoidance Methods for Urban Air Mobility through Simulation," Nov. 29, 2023, *arXiv*: arXiv:2311.18047. doi: 10.48550/arXiv.2311.18047.
[262] K. M. Chaturmutha and D. J. Mathew, "Factors Influencing the Exterior Design of Autonomous Passenger Drones: Literature Review," *Proc. Des. Soc.*, vol. 2, pp. 2087–2096, May 2022, doi: 10.1017/pds.2022.211.

[263] F. I. Anaroua and R. F. Babiceanu, “Airspace Integration of New Entrants and Safety Risk Models,” in *2024 IEEE International Conference on Automation, Quality and Testing, Robotics (AQTR)*, May 2024, pp. 1–6. doi: 10.1109/AQTR61889.2024.10554154.
[264] A. Andreeva-Mori, “Balancing predictability and flexibility through operation volume-constrained visual flight rule operations in low altitude airspaces,” *Front. Aerosp. Eng.*, vol. 3, Mar. 2024, doi: 10.3389/fpace.2024.1338388.
[265] V. Hopkin, “Real-time simulation in air traffic control,” *The Controller*, vol. 17, no. 4, 1978.
[266] D.-R. Schmitt, S. Kaltenhäuser, and B. Keck, “Real time simulation of integration of UAVs into Airspace,” in *26th congress of international council of the aeronautical sciences*, International Council of the Aeronautical Sciences, 2008. Accessed: Feb. 20, 2025. [Online]. Available: https://elib.dlr.de/54610/
[267] M. J. Hochrainer and P. Schattovich, “Real-Time Hybrid Simulation of an Unmanned Aerial Vehicle,” in *Dynamics of Coupled Structures, Volume 4*, M. S. Allen, R. L. Mayes, and D. J. Rixen, Eds., Cham: Springer International Publishing, 2017, pp. 41–48. doi: 10.1007/978-3-319-54930-9_4.
[268] S. Mukhopadhyay, C. Schurgers, D. Panigrahi, and S. Dey, “Model-Based Techniques for Data Reliability in Wireless Sensor Networks,” *IEEE Trans. Mob. Comput.*, vol. 8, pp. 528–543, 2009, doi: 10.1109/TMC.2008.131.
[269] S. Zelinski, “Validating The Airspace Concept Evaluation System Using Real World Data,” in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, in Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 2005. doi: 10.2514/6.2005-6491.
[270] F. D. Ferrare, D. M. Baum, J. R. de A. Júnior, J. B. C. Júnior, and P. S. Cugnasca, “Urban Air Mobility (UAM): A Model Proposal based on Agents using Netlogo,” presented at the 11th International Conference on Simulation and Modeling Methodologies, Technologies and Applications, Jun. 2025, pp. 352–359. Accessed: Jun. 17, 2025. [Online]. Available: https://www.scitepress.org/Link.aspx?doi=10.5220/0010557203520359
[271] H. Azad, V. Mehta, F. Dadboud, M. Bolic, and I. Mantegh, “Air-to-air simulated drone dataset for AI-powered problems,” presented at the 2023 IEEE/AIAA 42nd Digital Avionics Systems Conference (DASC), IEEE, 2023, pp. 1–7.
[272] M. N. Dehnavi, S. A. Yazdian, and S. J. Sadjadi, “Evaluating effective criteria on customer satisfaction using the best-worst method and optimizing resource allocation, case study Iran aseman airlines,” *J. Air Transp. Manag.*, vol. 109, p. 102375, Jun. 2023, doi: 10.1016/j.jairtraman.2023.102375.
[273] M. J. Kochenderfer, J. E. Holland, and J. P. Chryssanthacopoulos, “Next-Generation Airborne Collision Avoidance System,” *Linc. Lab. J.*, vol. 19, no. 1, p. 17, 2012.
[274] S. Wijnands, A. Sharpans’ kykh, and K. Aly, “Generation of Synthetic Aircraft Landing Trajectories Using Generative Adversarial Networks,” 2024.
[275] T. Souanef, S. Al-Rubaye, A. Tsourdos, S. Ayo, and D. Panagiotakopoulos, “Digital Twin Development for the Airspace of the Future,” *Drones*, vol. 7, no. 7, Art. no. 7, Jul. 2023, doi: 10.3390/drones7070484.
[276] C. Huang, H. Zhang, Y. Yang, L. He, and Q. Luo, “Research on fast generation and simulation technology of air traffic flow in control sector,” in *2021 IEEE 21st International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, Dec. 2021, pp. 987–993. doi: 10.1109/QRS-C55045.2021.00149.
[277] Y.-X. Han, J.-W. Zhang, and X.-Q. Huang, “Method for optimal conflict-free aircraft trajectory generation,” *Aircr. Eng. Aerosp. Technol.*, vol. 92, no. 2, pp. 156–162, Oct. 2019, doi: 10.1108/AEAT-07-2018-0207.
[278] D. Gianazza and N. Durand, “Separating air traffic flows by allocating 3D-trajectories,” in *The 23rd Digital Avionics Systems Conference*, Oct. 2004, p. 2.D.4-21. doi: 10.1109/DASC.2004.1391275.

[279] M. Mallick, Y. Yan, and S. Arulampalam, "Aircraft 3D Trajectory Estimation with a Single Nonlinear Filter Using Two 2D Radars," in *2018 International Conference on Control, Automation and Information Sciences (ICCAIS)*, Oct. 2018, pp. 405–410. doi: 10.1109/ICCAIS.2018.8570535.
[280] A. Baier and A. C. Zimmer, "Advantages of stereoscopic 3D displays for conflict judgment performance in air-traffic-control: when and how to use 3D," presented at the HCI International 2014-Posters' Extended Abstracts: International Conference, HCI International 2014, Heraklion, Crete, Greece, June 22-27, 2014. Proceedings, Part II 16, Springer, 2014, pp. 609–614.
[281] W. Tian, H. Xu, Y. Guo, B. Hu, and Y. Yao, "Probabilistic En Route Sector Traffic Demand Prediction Based upon Statistical Analysis of Error Distribution Characteristics," *J. Adv. Transp.*, vol. 2018, no. 1, p. 8184513, 2018, doi: 10.1155/2018/8184513.
[282] Y. Matsuno and T. Tsuchiya, "Stochastic 4D trajectory optimization for aircraft conflict resolution," in *2014 IEEE Aerospace Conference*, Mar. 2014, pp. 1–10. doi: 10.1109/AERO.2014.6836275.
[283] D. G. Mazzotta, "Conflict resolution algorithms for optimal trajectories in presence of uncertainty," Politecnico di Torino, 2019.
[284] Y. Wu, K. H. Low, and X. Hu, "Trajectory-based flight scheduling for AirMetro in urban environments by conflict resolution," *Transp. Res. Part C Emerg. Technol.*, vol. 131, p. 103355, Oct. 2021, doi: 10.1016/j.trc.2021.103355.
[285] O. Rodionova, "Aircraft trajectory optimization in North Atlantic oceanic airspace," phdthesis, Université Paul Sabatier - Toulouse III, 2015. Accessed: Jun. 17, 2025. [Online]. Available: https://theses.hal.science/tel-01214990
[286] J. M. Henderson, "Collaborative En Route Airspace Management Considering Stochastic Demand, Capacity, and Weather Conditions," Virginia Tech, 2008. Accessed: Jun. 17, 2025. [Online]. Available: http://hdl.handle.net/10919/26536
[287] A. Srivastava, "Rapid assessment of air traffic impact of blocking airspaces: Integrated communications navigation and surveillance (ICNS) conference," presented at the 2018 Integrated Communications, Navigation, Surveillance Conference (ICNS), IEEE, 2018, pp. 1–18.
[288] D. Sui, K. Liu, and Q. Li, "Dynamic Prediction of Air Traffic Situation in Large-Scale Airspace," *Aerospace*, vol. 9, no. 10, Art. no. 10, Oct. 2022, doi: 10.3390/aerospace9100568.
[289] T. Lauderdale, "Predictive models of the national airspace system using bayesian networks," presented at the Intl Congress of the Aeronautical Sciences (ICAS), Citeseer, 2008.
[290] L. Vempati, H. Kaing, A. Kane, and J. Bonn, "Terminal Area Modeling in National Airspace System Performance Analysis Capability (NASPAC)," in *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, in Aviation Technology, Integration, and Operations (ATIO) Conferences. American Institute of Aeronautics and Astronautics, Sep. 2010. doi: 10.2514/6.2010-9378.
[291] N. Shen, "Modeling of Airline and Passenger Dynamics in the National Airspace System (NAS)," Virginia Tech, 2010. Accessed: Jun. 17, 2025. [Online]. Available: http://hdl.handle.net/10919/77267
[292] G. Couluris, G. Carr, L. Meyn, K. Roth, A. Dabrowski, and J. Phillips, "Terminal Area Modeling Complexity Alternatives in a NAS-wide Simulation," in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, in Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 2004. doi: 10.2514/6.2004-5435.
[293] H. S. L. Fan, "Simulation of aircraft movements in the terminal airspace," *Transp. Plan. Technol.*, vol. 13, no. 1, pp. 31–42, Jan. 1989, doi: 10.1080/03081068808717386.
[294] D. Long, D. Lee, J. Johnson, E. Gaier, and P. Kostiuk, "Modeling air traffic management technologies with a queuing network model of the national airspace system," 1999.
[295] S. S. Holdren, M. Z. Li, and J. Hoffman, "Adaptable Graph Networks for Air Traffic Analysis Applications," in *2023 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, Apr. 2023, pp. 1–8. doi: 10.1109/ICNS58246.2023.10124325.
[296] G. Gurtner *et al.*, "Multi-Scale Analysis of the European Airspace Using Network Community Detection," *PLOS ONE*, vol. 9, no. 5, p. e94414, May 2014, doi: 10.1371/journal.pone.0094414.

[297] A. A. Baklanov and R. B. Nuterman, “Multi-scale atmospheric environment modelling for urban areas,” *Adv. Sci. Res.*, vol. 3, no. 1, pp. 53–57, Apr. 2009, doi: 10.5194/asr-3-53-2009.
[298] W. Wang and J. Xu, “A Tool for 3D Visualizing Moving Objects,” in *Web and Big Data*, L. Chen, C. S. Jensen, C. Shahabi, X. Yang, and X. Lian, Eds., Cham: Springer International Publishing, 2017, pp. 353–357. doi: 10.1007/978-3-319-63564-4_35.
[299] A. Debiasi, F. Prandi, G. Conti, R. D. Amicis, and R. Stojanovic, “Visual Analytics Tool For Urban Traffic Simulation,” presented at the Sixth International Conference on Simulation Tools and Techniques, Jul. 2013. Accessed: Jun. 17, 2025. [Online]. Available: https://eudl.eu/doi/10.4108/icst.simutools.2013.251740
[300] A. David and C. J. Tauro, “Web 3D data visualization of spatio temporal data using data driven document (D3js),” *Int. J. Comput. Appl.*, vol. 111, no. 4, 2015.
[301] A. Crowell, A. Fabian, and N. Nelson, “An interactive 4D visualization system for air traffic concept analysis,” in *2012 IEEE/AIAA 31st Digital Avionics Systems Conference (DASC)*, Oct. 2012, pp. 9E4-1-9E4-14. doi: 10.1109/DASC.2012.6382938.
[302] A. Cui, “A methodology for the design and development of an aerospace-specific data repository to support data-driven research,” Master of Applied Science, Bachelor’s Thesis, Institute of Informatics, Masaryk University, Brno, Kingston, Ontario, 2022.
[303] B. Wang *et al.*, “FLApy: A Python package for evaluating the 3D light availability heterogeneity within forest communities,” *Methods Ecol. Evol.*, 2024, doi: 10.1111/2041-210x.14382.
[304] C. Haedrich, V. Petras, A. Petrasova, S. Blumentrath, and H. Mitasova, “Integrating GRASS GIS and Jupyter Notebooks to facilitate advanced geospatial modeling education,” *Trans. GIS*, vol. 27, no. 3, pp. 686–702, 2023, doi: 10.1111/tgis.13031.
[305] M. Denno and G. Lemay, “Visualizing and Analyzing ADCP and CFD Data Using Python and Other Open Source Tools,” 2018, Accessed: Apr. 10, 2025. [Online]. Available: https://consensus.app/papers/visualizing-and-analyzing-adcp-and-cfd-data-using-python-denno-lemay/830ab00019615cc887a0de4870bb0f80/
[306] M. McKenney, N. Nyalakonda, J. McEvers, and M. Shipton, “Pyspatiotemporalgeom: a python library for spatiotemporal types and operations,” *Proc. 24th ACM SIGSPATIAL Int. Conf. Adv. Geogr. Inf. Syst.*, 2016, doi: 10.1145/2996913.2996973.
[307] I. Fountoulakis and C. Evangelidis, “SSA2py: A High-Performance Python Implementation of the Source-Scanning Algorithm for Spatiotemporal Seismic Source Imaging,” *Seismol. Res. Lett.*, 2024, doi: 10.1785/0220230335.
[308] R. Hales, “A New Method and Python Toolkit for General Access to Spatiotemporal N-Dimensional Raster Data,” 2021, Accessed: Apr. 10, 2025. [Online]. Available: https://consensus.app/papers/a-new-method-and-python-toolkit-for-general-access-to-hales/ae317e0fd796555e870c3df1da32cc93/
[309] P. S. Sommer, “Psyplot: Interactive data analysis and visualization with Python,” presented at the EGU General Assembly Conference Abstracts, 2018.
[310] M. Sergeeva, D. Delahaye, C. Mancel, and A. Vidosavljevic, “Dynamic airspace configuration by genetic algorithm,” *J. Traffic Transp. Eng. Engl. Ed.*, vol. 4, no. 3, pp. 300–314, Jun. 2017, doi: 10.1016/j.jtte.2017.05.002.
[311] M. Savai, J. Li, T. Wang, and I. Hwang, “An Algorithm for Adaptable Dynamic Airspace Configuration,” in *10th AIAA Aviation Technology, Integration, and Operations (ATIO) Conference*, in Aviation Technology, Integration, and Operations (ATIO) Conferences. American Institute of Aeronautics and Astronautics, Sep. 2010. doi: 10.2514/6.2010-9296.
[312] S. N. Hasan and J. A. Rossiter, “Sustaining trajectory flexibility for air traffic complexity alleviation,” in *AIP Conference Proceedings*, Nov. 2012, pp. 451–458. doi: 10.1063/1.4765528.
[313] V. J. Sciandra, J. Wei, I. Hwang, and W. D. Hall, “Benefit Analysis of a Sector Design Algorithm for Terminal Dynamic Airspace Configuration,” in *2013 Aviation Technology, Integration, and Operations Conference*, in AIAA AVIATION Forum. American Institute of Aeronautics and Astronautics, Aug. 2013. doi: 10.2514/6.2013-4261.

[314] K. Mehadhebi, “Operational risk assessment for airspace planning,” presented at the Proceedings of 7th USA/Europe Air Traffic Management Research and Development Seminar. Barcelona, Spain, 2007.
[315] U. Durak, “Special Issue: Reusability, Interoperability and Composability in Air Warfare Simulations,” *J. Def. Model. Simul.*, vol. 9, no. 2, pp. 99–100, Apr. 2012, doi: 10.1177/1548512912437648.
[316] G. L. Donohue, R. Holland, M. Luo, and C.-H. Chen, “Modeling the Performance of the National Airspace System,” *Transp. Res. Circ.*, 2002.
[317] M. Burgman, H. Layman, and S. French, “Eliciting Model Structures for Multivariate Probabilistic Risk Analysis,” *Front. Appl. Math. Stat.*, vol. 7, Jun. 2021, doi: 10.3389/fams.2021.668037.
[318] S. M. Katz, A.-C. Le Bihan, and M. J. Kochenderfer, “Learning an Urban Air Mobility Encounter Model from Expert Preferences,” in *2019 IEEE/AIAA 38th Digital Avionics Systems Conference (DASC)*, Sep. 2019, pp. 1–8. doi: 10.1109/DASC43569.2019.9081648.
[319] M. Smirti and M. Hansen, “Achieving a Higher Capacity National Airspace System: An Analysis of the Virtual Airspace Modeling and Simulation Project,” in *AIAA Modeling and Simulation Technologies Conference and Exhibit*, in Guidance, Navigation, and Control and Co-located Conferences. American Institute of Aeronautics and Astronautics, Aug. 2007. doi: 10.2514/6.2007-6698.
[320] M. I. Jordan and T. M. Mitchell, “Machine learning: Trends, perspectives, and prospects,” *Science*, vol. 349, no. 6245, pp. 255–260, 2015.
[321] T. M. Mitchell, *Machine Learning*. McGraw-Hill, 1997.
[322] M. Gottam, “How Machine Learning Can Be Used To Improve Predictive Analytics,” *Int. J. Res. Appl. Sci. Eng. Technol.*, 2022, doi: 10.22214/ijraset.2022.48432.
[323] J. G. Richens, C. M. Lee, and S. Johri, “Improving the accuracy of medical diagnosis with causal machine learning,” *Nat. Commun.*, vol. 11, 2020, doi: 10.1038/s41467-020-17419-7.
[324] M. Loukili, F. Messaoudi, and M. E. Ghazi, “Personalizing Product Recommendations using Collaborative Filtering in Online Retail: A Machine Learning Approach,” *2023 Int. Conf. Inf. Technol. ICIT*, pp. 19–24, 2023, doi: 10.1109/ICIT58056.2023.10226042.
[325] I. Goodfellow, Y. Bengio, and A. Courville, *Deep Learning*. MIT Press, 2016. [Online]. Available: http://www.deeplearningbook.org
[326] J. A. M. Sidey-Gibbons and C. Sidey-Gibbons, “Machine learning in medicine: a practical introduction,” *BMC Med. Res. Methodol.*, vol. 19, 2019, doi: 10.1186/s12874-019-0681-4.
[327] C. M. Bishop, *Pattern Recognition and Machine Learning*, 2006. Corr. 2nd Printing 2011 ed. edition. New York: Springer/Sci-Tech/Trade, 2006.
[328] T. Hastie, R. Tibshirani, J. H. Friedman, and J. H. Friedman, *The elements of statistical learning: data mining, inference, and prediction*, vol. 2. Springer, 2009.
[329] X. Zhu and A. B. Goldberg, *Introduction to semi-supervised learning*. Springer Nature, 2022.
[330] R. S. Sutton, “Introduction: The Challenge of Reinforcement Learning,” *Mach. Learn.*, 1992, doi: 10.1007/BF00992695.
[331] M. L. Littman, “Reinforcement learning improves behaviour from evaluative feedback,” *Nature*, vol. 521, no. 7553, pp. 445–451, May 2015, doi: 10.1038/nature14540.
[332] A. Gupta, A. Sharma, and A. Goel, “Review of regression analysis models,” *Int J Eng Res Technol*, vol. 6, no. 08, pp. 58–61, 2017.
[333] J. Chen, Y. Zhang, and J. Hu, “Analysis of Flight Fuel Consumption Based on Nonlinear Regression,” *DEStech Trans. Comput. Sci. Eng.*, 2017, doi: 10.12783/DTCSE/CST2017/12570.
[334] R. Raman, R. Chaubey, S. Goswami, and R. Jella, “Experiences in Machine Learning Models for Aircraft Fuel Flow and Drag Polar Predictions,” *Stud. Comput. Intell.*, 2021, doi: 10.1007/978-3-030-68291-0_27.
[335] A. S. A. Al-Khanbashi and C. Jing, “Data-driven Fuel Flow Prediction Model for Aircraft Engines,” *Proc. 2023 3rd Int. Conf. Big Data Artif. Intell. Risk Manag.*, 2023, doi: 10.1145/3656766.3656814.

[336] K. Hu, Y. Cheng, J. Wu, H. Zhu, and X. Shao, "Deep Bidirectional Recurrent Neural Networks Ensemble for Remaining Useful Life Prediction of Aircraft Engine," *IEEE Trans. Cybern.*, vol. 53, pp. 2531–2543, 2021, doi: 10.1109/TCYB.2021.3124838.
[337] D. Pebrianti, Z. Khalani, Rusdah, and L. Bayuaji, "Predictive Maintenance in Aerospace Industry Using Convolutional Neural Network," *2024 9th Int. Conf. Mechatron. Eng. ICOM*, pp. 157–162, 2024, doi: 10.1109/ICOM61675.2024.10652505.
[338] S. Sharma, A. K. Pandit, and S. S, "Predicting Aircraft Turbofan Engine Degradation with Recurrent Neural Networks," *2024 IEEE Int. Conf. Inf. Technol. Electron. Intell. Commun. Syst. ICITEICS*, pp. 1–6, 2024, doi: 10.1109/ICITEICS61368.2024.10625502.
[339] Y. Zhu, Z. Liu, Z. Luo, C. Du, and H. Wang, "Aircraft engine remaining life prediction method with deep learning," *2022 Int. Conf. Artif. Intell. Comput. Inf. Technol. AICIT*, pp. 1–6, 2022, doi: 10.1109/AICIT55386.2022.9930216.
[340] S. B. Kotsiantis, I. Zaharakis, and P. Pintelas, "Supervised machine learning: A review of classification techniques," *Emerg. Artif. Intell. Appl. Comput. Eng.*, vol. 160, no. 1, pp. 3–24, 2007.
[341] C. S. Bosson and T. Nikoleris, "Supervised Learning Applied to Air Traffic Trajectory Classification," in *2018 AIAA Information Systems-AIAA Infotech @ Aerospace*, in AIAA SciTech Forum. American Institute of Aeronautics and Astronautics, Jan. 2018. doi: 10.2514/6.2018-1637.
[342] J. A. Pérez-Castán, L. Pérez-Sanz, L. Serrano-Mira, F. J. Saéz-Hernando, I. Rodríguez Gauxachs, and V. F. Gómez-Comendador, "Design of an ATC Tool for Conflict Detection Based on Machine Learning Techniques," *Aerospace*, vol. 9, no. 2, Art. no. 2, Feb. 2022, doi: 10.3390/aerospace9020067.
[343] D. Truong and W. Choi, "Using machine learning algorithms to predict the risk of small Unmanned Aircraft System violations in the National Airspace System," *J. Air Transp. Manag.*, vol. 86, p. 101822, Jul. 2020, doi: 10.1016/j.jairtraman.2020.101822.
[344] O. Okun, G. Valentini, and M. Re, *Ensembles in machine learning applications*, vol. 373. Springer Science & Business Media, 2011.
[345] I. D. Mienye and Y. Sun, "A Survey of Ensemble Learning: Concepts, Algorithms, Applications, and Prospects," *IEEE Access*, vol. 10, pp. 99129–99149, 2022, doi: 10.1109/ACCESS.2022.3207287.
[346] S. Papadopoulos, Azar ,Elie, Woon ,Wei-Lee, and C. E. and Kontokosta, "Evaluation of tree-based ensemble learning algorithms for building energy performance estimation," *J. Build. Perform. Simul.*, vol. 11, no. 3, pp. 322–332, May 2018, doi: 10.1080/19401493.2017.1354919.
[347] D. Ziyadullaev, D. Muhamediyeva, K. Madazimov, M. Madazimov, P. Temirov, and D. Abdukadirov, "Application of ensemble machine learning methods for diabetes diagnosis," *BIO Web Conf.*, vol. 121, p. 01002, 2024, doi: 10.1051/bioconf/202412101002.
[348] M. M. Gana, M. Ismail, and A. M. Mabu, "Credit Card Fraud Detection using Ensemble Learning with Boosting Technique," *Int. J. Sci. Dev. Res.*, vol. 8, no. 8, pp. 1170–1179, Aug. 2023.
[349] S. Y. Prasetyo, "Synergizing Ensemble Techniques for Precision Breast Cancer Diagnosis: A Comparative Performance Analysis," in *2024 IEEE 10th International Conference on Smart Instrumentation, Measurement and Applications (ICSIMA)*, Jul. 2024, pp. 297–302. doi: 10.1109/ICSIMA62563.2024.10675541.
[350] H. T. Hoc, R. Silhavy, Z. Prokopova, and P. Silhavy, "Comparing Stacking Ensemble and Deep Learning for Software Project Effort Estimation," *IEEE Access*, vol. 11, pp. 60590–60604, 2023, doi: 10.1109/ACCESS.2023.3286372.
[351] W. Zhou, D.-T. Pham, and S. Alam, "AirFusion: A Machine Learning Framework for Balancing Air Traffic Demand and Airspace Capacity Through Dynamic Airspace Sectorization," in *2023 IEEE 26th International Conference on Intelligent Transportation Systems (ITSC)*, Sep. 2023, pp. 5324–5331. doi: 10.1109/ITSC57777.2023.10421895.
[352] Y.-E. Kang and Y.-H. Jung, "Machine Learning-Based Air-to-Ground Channel Model Selection Method for UAV Communications Using Digital Surface Model Data," *Sensors*, vol. 22, no. 23, Art. no. 23, Jan. 2022, doi: 10.3390/s22239234.

[353] W. Zixuan, Z. Ning, H. Weijun, and Y. Sheng, "Study on Prediction Method of Flight Fuel Consumption with Machine Learning," in *2020 IEEE International Conference on Information Technology,Big Data and Artificial Intelligence (ICIBA)*, Nov. 2020, pp. 624–627. doi: 10.1109/ICIBA50161.2020.9277445.
[354] R. Zhang, T. Du, and S. Qu, "A Principal Component Analysis Algorithm Based on Dimension Reduction Window," *IEEE Access*, vol. 6, pp. 63737–63747, 2018, doi: 10.1109/ACCESS.2018.2875270.
[355] D. S. Karlov, S. Sosnin, I. V. Tetko, and M. V. Fedorov, "Chemical space exploration guided by deep neural networks," *RSC Adv.*, vol. 9, no. 9, pp. 5151–5157, Feb. 2019, doi: 10.1039/C8RA10182E.
[356] J. Li and Y. Wang, "nPCA: a linear dimensionality reduction method using a multilayer perceptron," *Front. Genet.*, vol. 14, Jan. 2024, doi: 10.3389/fgene.2023.1290447.
[357] H. Navya, "Influence over the Dimensionality Reduction and Clustering for Air Quality Measurements using PCA and SOM," *Int. J. Adv. Eng. Manag. Sci.*, vol. 3, no. 11, p. 239936.
[358] N. Migenda, R. Möller, and W. Schenck, "Adaptive dimensionality reduction for neural network-based online principal component analysis," *PLOS ONE*, vol. 16, no. 3, p. e0248896, Mar. 2021, doi: 10.1371/journal.pone.0248896.
[359] et al U. Gupta, "A Unique Pipeline Model to Improve Anomaly Detection in High Dimensional Data," *Int. J. Recent Innov. Trends Comput. Commun.*, vol. 11, no. 10, Art. no. 10, Nov. 2023, doi: 10.17762/ijritcc.v11i10.8810.
[360] H. Jafarzadeh, R. R. Torkashvand, C. Asgari, and and A. Amiry, "Provide A New Approach for Mining Fuzzy Association Rules using Apriori Algorithm," *Indian J. Sci. Technol.*, vol. 8, no. Supplementary 7, pp. 1–8, Apr. 2015, doi: 10.17485/ijst/2015/v8iS7/71227.
[361] O. T. Matthew and M. M. Lawal, "FP-Growth Algorithm: Mining Association Rules without Candidate Sets Generation," *Kasu J. Comput. Sci.*, 2024, doi: 10.47514/kjcs/2024.1.2.0016.
[362] V. Rajpoot, "An Enhanced Eclat Algorithm for Mining Association Rules using data mining," 2018. Accessed: Jun. 17, 2025. [Online]. Available: https://www.semanticscholar.org/paper/An-Enhanced-Eclat-Algorithm-for-Mining-Association-Rajpoot/14efd5de3a7ab0757690812f0ba0eaf9ee899cd4?utm_source=consensus
[363] M. Ganjir and J. Chopra, "Combining Apriori and FP Growth algorithms with Simulated Annealing for Optimized Association Rule Mining," 2016. Accessed: Jul. 21, 2026. [Online]. Available: https://www.semanticscholar.org/paper/Combining-Apriori-and-FP-Growth-algorithms-with-for-Ganjir-Chopra/d88b4700b048c582f7463dafaf3c4798dead1ee8
[364] Nguyen H. D. and Tran T. D., "Detecting outliers in GNSS position time series using machine learning techniques," *J. Min. Earth Sci.*, vol. 64, no. 4, Art. no. 4, Aug. 2023, doi: 10.46326/JMES.2023.64(4).03.
[365] P. Russell, M. A. Elsayed, B. Nandy, N. Seddigh, and N. Zincir-Heywood, "On the Fence: Anomaly Detection in IoT Networks," in *NOMS 2023-2023 IEEE/IFIP Network Operations and Management Symposium*, May 2023, pp. 1–4. doi: 10.1109/NOMS56928.2023.10154271.
[366] H. Bilakanti, S. Pasam, V. Palakollu, and S. Utukuru, "Anomaly detection in IoT environment using machine learning," *Secur. Priv.*, vol. 7, no. 3, p. e366, 2024, doi: 10.1002/spy2.366.
[367] R. Rawat, K. AL-Attab, K. K. Dixit, A. Deepak, G. Pushkarna, and M. Harikrishna, "Real-Time Anomaly Detection in Large-Scale Sensor Networks using Isolation Forests," in *2024 International Conference on Communication, Computer Sciences and Engineering (IC3SE)*, May 2024, pp. 1400–1405. doi: 10.1109/IC3SE62002.2024.10593443.
[368] C. Yang, Y. Wang, Y. Zhou, J. Ruan, and W. Liu, "False Data Injection Attacks Detection in Power System Using Machine Learning Method," *J. Comput. Commun.*, vol. 6, no. 11, Art. no. 11, Nov. 2018, doi: 10.4236/jcc.2018.611025.
[369] S. Shriram and E. Sivasankar, "Anomaly Detection on Shuttle data using Unsupervised Learning Techniques," in *2019 International Conference on Computational Intelligence and Knowledge Economy (ICCIKE)*, Dec. 2019, pp. 221–225. doi: 10.1109/ICCIKE47802.2019.9004325.
[370] S. Velunachiyar and K. Sivakumar, "Some Clustering Methods, Algorithms and their Applications".

[371] S. Chander and P. Vijaya, “Unsupervised learning methods for data clustering,” pp. 41–64, 2021, doi: 10.1016/B978-0-12-820601-0.00002-1.
[372] X. Sáez-de-Cámara, J. L. Flores, C. Arellano, A. Urbieta, and U. Zurutuza, “Clustered federated learning architecture for network anomaly detection in large scale heterogeneous IoT networks,” *Comput. Secur.*, vol. 131, p. 103299, Aug. 2023, doi: 10.1016/j.cose.2023.103299.
[373] F. Nicol and S. Puechmorel, “Minimum entropy unsupervised aircraft trajectories clustering: theory and implementation,” *Int. J. Adv. Softw.*, vol. 9, no. 3–4, 2016, Accessed: Jun. 18, 2025. [Online]. Available: https://enac.hal.science/hal-01367593
[374] S. J. Corrado, T. G. Puranik, O. J. Pinon, and D. N. Mavris, “Trajectory Clustering within the Terminal Airspace Utilizing a Weighted Distance Function,” *Proceedings*, vol. 59, no. 1, Art. no. 1, 2020, doi: 10.3390/proceedings2020059007.
[375] F. Nicol and S. Puechmorel, “Unsupervised aircraft trajectories clustering: a minimum entropy approach,” presented at the ALLDATA 2016, 2nd International Conference on Big Data, Small Data, Linked Data and Open Data, 2016, p. ISBN-978.
[376] S. Lee and Y. Lee, “A Comparative Analysis of Clustering Algorithms for Assessing College Students’ Key Competencies,” *J. Educ. Eval.*, vol. 37, no. 3, pp. 411–439, Sep. 2024, doi: 10.31158/JEEV.2024.37.3.411.
[377] Praveena Kumari, M. D. H, and A. K. M, “Clustering Algorithms to Analyse Smart City Traffic Data,” *Int. J. Adv. Comput. Sci. Appl. Ijacsa*, vol. 15, no. 8, Art. no. 8, Jul. 2024, doi: 10.14569/IJACSA.2024.0150811.
[378] J. Q. Jerin, N. K. Khan, S. Biswas, and N. Sharmin, “Comparative Study of Clustering Algorithms: Scenario Based on Boston Crime Dataset,” in *2023 26th International Conference on Computer and Information Technology (ICCIT)*, Dec. 2023, pp. 1–6. doi: 10.1109/ICCIT60459.2023.10441641.
[379] Sravana Kumar, Sreenivasa Rao, and P. Srinivasrao, “An Efficient Clustering Approach using DBSCAN – Helix,” *Helix*, vol. 8, no. 3, 2018, doi: DOI 10.29042/2018-3399-3405.
[380] S. K. Jasra, G. Valentino, A. Muscat, and R. Camilleri, “Hybrid Machine Learning–Statistical Method for Anomaly Detection in Flight Data,” *Appl. Sci.*, vol. 12, no. 20, Art. no. 20, Jan. 2022, doi: 10.3390/app122010261.
[381] X. Olive and L. Basora, “Identifying Anomalies in past en-route Trajectories with Clustering and Anomaly Detection Methods,” in *ATM Seminar 2019*, Vienne, Austria, Jun. 2019. Accessed: Jun. 18, 2025. [Online]. Available: https://hal.science/hal-02345597
[382] A. Chandra, S. Hazra, A. Verma, and K. P. Sooraj, “On the Possibilities of Efficient Air Traffic Monitoring through Complex Network Clustering Based Airspace Sub-Sectorization: A Multi-Objective Discrete Particle Swarm Optimization Approach,” *Transp. Res. Rec.*, vol. 2679, no. 2, pp. 2127–2143, Feb. 2025, doi: 10.1177/03611981241263829.
[383] Z. Wang, M. Liang, and D. Delahaye, “A hybrid machine learning model for short-term estimated time of arrival prediction in terminal manoeuvring area,” *Transp. Res. Part C Emerg. Technol.*, vol. 95, pp. 280–294, Oct. 2018, doi: 10.1016/j.trc.2018.07.019.
[384] A. K. Jain, “Data clustering: 50 years beyond K-means,” *Pattern Recognit. Lett.*, vol. 31, no. 8, pp. 651–666, Jun. 2010, doi: 10.1016/j.patrec.2009.09.011.
[385] T. Kanungo, D. M. Mount, N. S. Netanyahu, C. D. Piatko, R. Silverman, and A. Y. Wu, “An efficient k-means clustering algorithm: analysis and implementation,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 24, no. 7, pp. 881–892, Jul. 2002, doi: 10.1109/TPAMI.2002.1017616.
[386] T. Bilen, P. J. Aydemir, A. E. Konu, and B. Canberk, “Customized K-Means Based Topology Clustering for Aeronautical Ad-hoc Networks,” *2021 IEEE 26th Int. Workshop Comput. Aided Model. Des. Commun. Links Netw. CAMAD*, pp. 1–5, 2021, doi: 10.1109/camad52502.2021.9617810.
[387] A. M. Baswade and P. S. Nalwade, “Selection of Initial Centroids for k-Means Algorithm,” 2013, Accessed: Dec. 12, 2024. [Online]. Available: https://consensus.app/papers/selection-of-initial-centroids-for-kmeans-algorithm-baswade-nalwade/eec974cf02ff5b4b9805d72eeae1a00f/
[388] Y. Cheung, “k*-Means: A new generalized k-means clustering algorithm,” *Pattern Recognit Lett*, vol. 24, pp. 2883–2893, 2003, doi: 10.1016/S0167-8655(03)00146-6.

[389] C. Shi, B. Wei, S. Wei, W. Wang, H. Liu, and J. Liu, "A quantitative discriminant method of elbow point for the optimal number of clusters in clustering algorithm," *EURASIP J. Wirel. Commun. Netw.*, vol. 2021, no. 1, p. 31, Feb. 2021, doi: 10.1186/s13638-021-01910-w.
[390] H. Zhong, H. Liu, and G. Qi, "Analysis of Terminal Area Airspace Operation Status Based on Trajectory Characteristic Point Clustering," *IEEE Access*, vol. 9, pp. 16642–16648, 2021, doi: 10.1109/ACCESS.2021.3053012.
[391] P. Govender and V. Sivakumar, "Application of k-means and hierarchical clustering techniques for analysis of air pollution: A review (1980–2019)," *Atmos. Pollut. Res.*, vol. 11, pp. 40–56, 2020, doi: 10.1016/j.apr.2019.09.009.
[392] X. Tang, J. Gu, Z. Shen, and P. Chen, "A flight profile clustering method combining twed with K-means algorithm for 4D trajectory prediction," in *2015 Integrated Communication, Navigation and Surveillance Conference (ICNS)*, Apr. 2015, pp. S3-1-S3-9. doi: 10.1109/ICNSURV.2015.7121260.
[393] M. Ester *et al.*, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proceedings of the Second International Conference on Knowledge Discovery and Data Mining*, in Guide Proceedings. Aug. 1996, pp. 226–231. doi: 10.5555/3001460.3001507.
[394] E. Schubert, J. Sander, M. Ester, H. P. Kriegel, and X. Xu, "DBSCAN revisited, revisited: why and how you should (still) use DBSCAN," *ACM Trans. Database Syst. TODS*, vol. 42, no. 3, pp. 1–21, 2017, doi: https://doi-org.journal.rmc.ca/10.1145/3068335.
[395] Y. Zhenxing, "An Improved DBSCAN Algorithm," *J. Sichuan Norm. Univ.*, 2013, Accessed: Dec. 12, 2024. [Online]. Available: https://consensus.app/papers/an-improved-dbscan-algorithm-zhenxing/55490eba4b72500babe3fc1be29e5f25/
[396] T. Tran, K. Drab, and M. Daszykowski, "Revised DBSCAN algorithm to cluster data with dense adjacent clusters," *Chemom. Intell. Lab. Syst.*, vol. 120, pp. 92–96, 2013, doi: 10.1016/J.CHEMOLAB.2012.11.006.
[397] S. Bhardwaj and S. Dash, "VDMR-DBSCAN: Varied Density MapReduce DBSCAN," pp. 134–150, 2015, doi: 10.1007/978-3-319-27057-9_10.
[398] K. Ahmed and T. Razak, "An Overview of Various Improvements of DBSCAN Algorithm in Clustering Spatial Databases," 2016, Accessed: Dec. 12, 2024. [Online]. Available: https://consensus.app/papers/an-overview-of-various-improvements-of-dbscan-algorithm-in-ahmed-razak/25979d0f1eba51dea532b02faa38b407/
[399] K. Khan, S. U. Rehman, K. Aziz, S. Fong, and S. Sarasvady, "DBSCAN: Past, present and future," presented at the The fifth international conference on the applications of digital information and web technologies (ICADIWT 2014), IEEE, 2014, pp. 232–238. doi: 10.1109/ICADIWT.2014.6814687.
[400] T. Tran, T.-T. Nguyen, T. A. Willemsz, G. van Kessel, H. Frijlink, and K. van der V. Maarschalk, "A density-based segmentation for 3D images, an application for X-ray micro-tomography.," *Anal. Chim. Acta*, vol. 725, pp. 14–21, 2012, doi: 10.1016/j.aca.2012.03.008.
[401] N. Saada, T. Harsono, and A. Basuki, "Improvement of Segmentation Performance for Feature Extraction on Whirlwind Cloud-based Satellite Image using DBSCAN Clustering Algorithm," *Emit. Int. J. Eng. Technol.*, 2019, doi: 10.24003/EMITTER.V7I1.372.
[402] C. E. V. Gallego, V. F. Gómez Comendador, F. J. Saez Nieto, and M. G. Martinez, "Discussion On Density-Based Clustering Methods Applied for Automated Identification of Airspace Flows," in *2018 IEEE/AIAA 37th Digital Avionics Systems Conference (DASC)*, Sep. 2018, pp. 1–10. doi: 10.1109/DASC.2018.8569219.
[403] R. Deshmukh, "Data-Driven Anomaly and Precursor Detection in Metroplex Airspace Operations," thesis, Purdue University Graduate School, 2020. doi: 10.25394/PGS.12121095.v1.
[404] D. Comaniciu and P. Meer, "Mean Shift: A Robust Approach Toward Feature Space Analysis," *IEEE Trans Pattern Anal Mach Intell*, vol. 24, pp. 603–619, 2002, doi: 10.1109/34.1000236.
[405] Y.-Z. Ren, U. Kamath, C. Domeniconi, and G. Zhang, "Boosted Mean Shift Clustering," pp. 646–661, 2014, doi: 10.1007/978-3-662-44851-9_41.
[406] O. Tuzel, F. Porikli, and P. Meer, "Kernel methods for weakly supervised mean shift clustering," *2009 IEEE 12th Int. Conf. Comput. Vis.*, pp. 48–55, 2009, doi: 10.1109/ICCV.2009.5459204.

[407] O. Hyrien and A. Baran, “Fast Nonparametric Density-Based Clustering of Large Datasets Using a Stochastic Approximation Mean-Shift Algorithm,” *J. Comput. Graph. Stat.*, vol. 25, pp. 899–916, 2016, doi: 10.1080/10618600.2015.1051625.
[408] C. Bepery, S. Bhadra, M. M. Rahman, M. Sarkar, and M. J. Hossain, “Improved Mean Shift Algorithm for Maximizing Clustering Accuracy,” *J. Eng. Adv.*, 2021, doi: 10.38032/jea.2021.01.001.
[409] B. Georgescu, I. Shimshoni, and P. Meer, “Mean shift based clustering in high dimensions: a texture classification example,” *Proc. Ninth IEEE Int. Conf. Comput. Vis.*, pp. 456–4631, 2003, doi: 10.1109/ICCV.2003.1238382.
[410] Y. Peng, X. Wen, J. Kong, Y. Meng, and M. Wu, “A Study on the Normalized Delineation of Airspace Sectors Based on Flight Conflict Dynamics,” *Appl. Sci.*, vol. 13, no. 21, Art. no. 21, Jan. 2023, doi: 10.3390/app132112070.
[411] B. Levy, “Track averages and traffic pattern characterization for airspace analysis,” *Digit. Avion. Syst. Conf. 2003 DASC 03 22nd*, vol. 1, pp. 14–11, 2003, doi: 10.1109/DASC.2003.1245811.
[412] R. Zhu, C. Wang, S. Jin, J. Zhang, M. Lu, and H. Li, “Radar Signal Pre-sorting Based on Mean-Shift Algorithm,” *2022 7th Int. Conf. Intell. Comput. Signal Process. ICSP*, pp. 1766–1769, 2022, doi: 10.1109/ICSP54964.2022.9778459.
[413] J. Li and I. Hwang, “Algorithms for Uncertainty Analysis of an Air Traffic Management System,” 2008, doi: 10.2514/6.2008-6328.
[414] X. Zhuang, Y. Huang, K. Palaniappan, and Y. Zhao, “Gaussian mixture density modeling, decomposition, and applications,” *IEEE Trans. Image Process.*, vol. 5, no. 9, pp. 1293–1302, 1996.
[415] F. Nielsen, V. Garcia, and R. Nock, “Simplifying Gaussian mixture models via entropic quantization,” presented at the 2009 17th European Signal Processing Conference, IEEE, 2009, pp. 2012–2016.
[416] D. Peel and G. MacLahlan, *Finite mixture models*. John Wiley & Sons, Inc., 2000.
[417] H. Nguyen and G. McLachlan, “Maximum likelihood estimation of Gaussian mixture models without matrix operations,” *Adv. Data Anal. Classif.*, vol. 9, pp. 371–394, 2015, doi: 10.1007/s11634-015-0209-7.
[418] P. D. McNicholas and T. B. Murphy, “Parsimonious Gaussian mixture models,” *Stat. Comput.*, vol. 18, no. 3, pp. 285–296, Sep. 2008, doi: 10.1007/s11222-008-9056-0.
[419] C. Biernacki, G. Celeux, and G. Govaert, “Assessing a Mixture Model for Clustering with the Integrated Completed Likelihood,” *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 7, pp. 719–725, Jul. 2000, doi: 10.1109/34.865189.
[420] T. Xiong, L. Zhang, and Z. Yi, “Double Gaussian mixture model for image segmentation with spatial relationships,” *J. Vis. Commun. Image Represent.*, vol. 34, pp. 135–145, Jan. 2016, doi: 10.1016/j.jvcir.2015.10.018.
[421] J. Samuelsson, “Waveform quantization of speech using Gaussian mixture models,” *2004 IEEE Int. Conf. Acoust. Speech Signal Process.*, vol. 1, 2004, doi: 10.1109/ICASSP.2004.1325948.
[422] J. Krause, “Mixture models in financial risk modeling,” Jan. 2014.
[423] W. Zeng, Z. Xu, Z. Cai, X. Chu, and X. Lu, “Aircraft Trajectory Clustering in Terminal Airspace Based on Deep Autoencoder and Gaussian Mixture Model,” *Aerospace*, vol. 8, no. 9, Art. no. 9, Sep. 2021, doi: 10.3390/aerospace8090266.
[424] W. Wu, J. M. Keller, M. Skubic, and M. Popescu, “Data Stream Trajectory Analysis Using Sequential Possibilistic Gaussian Mixture Model,” in *2019 IEEE International Conference on Fuzzy Systems (FUZZ-IEEE)*, Jun. 2019, pp. 1–7. doi: 10.1109/FUZZ-IEEE.2019.8858874.
[425] H. Paek, K. Lee, and A. Vela, “En-route Arrival Time Prediction using Gaussian Mixture Model,” 2020, Accessed: Dec. 12, 2024. [Online]. Available: https://consensus.app/papers/enroute-arrival-time-prediction-using-gaussian-mixture-paek-lee/e6772702ac775cf3845df3d2a62d82ee/
[426] A. Min, R. Sagarna, A. Gupta, Y. Ong, and C. Goh, “Knowledge Transfer Through Machine Learning in Aircraft Design,” *IEEE Comput. Intell. Mag.*, vol. 12, pp. 48–60, 2017, doi: 10.1109/MCI.2017.2742781.

[427] P. Yao, H. Wang, and H. Ji, "Gaussian mixture model and receding horizon control for multiple UAV search in complex environment," *Nonlinear Dyn.*, vol. 88, no. 2, pp. 903–919, Apr. 2017, doi: 10.1007/s11071-016-3284-1.
[428] W. Zhao, L. Li, S. Alam, and Y. Wang, "An incremental clustering method for anomaly detection in flight data," *Transp. Res. Part C Emerg. Technol.*, vol. 132, p. 103406, Nov. 2021, doi: 10.1016/j.trc.2021.103406.
[429] J. W. T. Lee, D. Yeung, and E. C. C. Tsang, "Hierarchical clustering based on ordinal consistency," *Pattern Recognit*, vol. 38, pp. 1913–1925, 2005, doi: 10.1016/j.patcog.2005.05.008.
[430] D. Dabhi and M. R. Patel, "Extensive Survey on Hierarchical Clustering Methods in Data Mining," 2016, Accessed: Dec. 12, 2024. [Online]. Available: https://consensus.app/papers/extensive-survey-on-hierarchical-clustering-methods-in-dabhi-patel/a777c4bfac40575199582266b998d542/
[431] H. M. Y. A. Dalbouh and N. Norwawi, "Improvement on Agglomerative Hierarchical Clustering Algorithm Based on Tree Data Structure with Bidirectional Approach," *2012 Third Int. Conf. Intell. Syst. Model. Simul.*, pp. 25–30, 2012, doi: 10.1109/ISMS.2012.13.
[432] Z. Nazari, M. Nazari, and D. Kang, "A bottom-up hierarchical clustering algorithm with intersection points," 2018, Accessed: Dec. 12, 2024. [Online]. Available: https://consensus.app/papers/a-bottomup-hierarchical-clustering-algorithm-with-nazari-nazari/93bc42231d585225ad730bbaf08a2f20/
[433] T. Zhang, R. Ramakrishnan, and M. Livny, "BIRCH: an efficient data clustering method for very large databases," in *Proceedings of the 1996 ACM SIGMOD international conference on Management of data*, in SIGMOD '96. New York, NY, USA: Association for Computing Machinery, Jun. 1996, pp. 103–114. doi: 10.1145/233269.233324.
[434] S. J. Russell and P. Norvig, *Artificial intelligence: a modern approach*. Englewood Cliffs, N.J.: Prentice Hall, 1995. doi: 10.1017/S0269888900007724.
[435] N. Godcares, A. Sirsath, A. Bongale, P. Kadam, R. Jayawal, and S. Patil, "Exploring Customer Segmentation in the Context of Market Analysis," *2023 IEEE 11th Reg. 10 Humanit. Technol. Conf. R10-HTC*, pp. 444–449, 2023, doi: 10.1109/R10-HTC57504.2023.10461815.
[436] J. Oyelade *et al.*, "Clustering Algorithms: Their Application to Gene Expression Data," *Bioinforma. Biol. Insights*, vol. 10, pp. 237–253, Nov. 2016, doi: 10.4137/BBI.S38316.
[437] V. Jabade, S. Ghadge, M. Jamadar, and P. Girase, "Customer Segmentation for Smooth Shopping Experience," *2023 4th Int. Conf. Emerg. Technol. INCET*, pp. 1–7, 2023, doi: 10.1109/INCET57972.2023.10170126.
[438] C. Deng, K. Kim, H.-C. Choi, and I. Hwang, "Trajectory Pattern Identification for Arrivals in Vectored Airspace," in *2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC)*, Oct. 2021, pp. 1–8. doi: 10.1109/DASC52595.2021.9594503.
[439] X. Chu, X. Tan, and W. Zeng, "A Clustering Ensemble Method of Aircraft Trajectory Based on the Similarity Matrix," *Aerospace*, vol. 9, no. 5, p. 269, May 2022, doi: 10.3390/aerospace9050269.
[440] J. Zhao and M. Sartipi, "Automatic Identification of Anomalous Driving Events from Trajectory Data," in *2022 IEEE 25th International Conference on Intelligent Transportation Systems (ITSC)*, Oct. 2022, pp. 851–856. doi: 10.1109/ITSC55140.2022.9922078.
[441] S. Saini and P. Rani, "A Survey on STING and CLIQUE Grid Based Clustering Methods," *Int. J. Adv. Res. Comput. Sci.*, vol. 8, no. 5, Art. no. 5, Jun. 2017, doi: 10.26483/ijarcs.v8i5.3616.
[442] H. He, Y. He, and C. Zhu, "An Improved CLIQUE Algorithm," *J. Phys. Conf. Ser.*, vol. 2400, no. 1, p. 012048, Dec. 2022, doi: 10.1088/1742-6596/2400/1/012048.
[443] S. Parmar, Rahul, and S. Taneja, "CONGEST: An Algorithm to Detect Congestion Zones Based on Unmanned Aerial Vehicle (UAV) Flight plans," in *2023 2nd International Conference on Vision Towards Emerging Trends in Communication and Networking Technologies (ViTECoN)*, May 2023, pp. 1–6. doi: 10.1109/ViTECoN58111.2023.10157846.
[444] M. Cai, L. Wan, Z. Jiao, M. Lv, Z. Gao, and D. Qi, "Clustering Method of Large-Scale Battlefield Airspace Based on Multi A * in Airspace Grid System," *Appl. Sci.*, vol. 12, no. 22, Art. no. 22, Jan. 2022, doi: 10.3390/app122211396.

[445] M. Y. Choong, L. Angeline, R. K. Y. Chin, K. B. Yeo, and K. T. K. Teo, "Modeling of Vehicle Trajectory using K-Means and Fuzzy C-Means Clustering," in *2018 IEEE International Conference on Artificial Intelligence in Engineering and Technology (IICAIET)*, Nov. 2018, pp. 1–6. doi: 10.1109/IICAIET.2018.8638471.
[446] Q. Chen, "Design of Target Detection Algorithm in Airport Flight Area Based on Fuzzy Clustering," in *2023 2nd International Conference on 3D Immersion, Interaction and Multi-sensory Experiences (ICDIIME)*, Jun. 2023, pp. 41–45. doi: 10.1109/ICDIIME59043.2023.00014.
[447] M. V. Akinin, N. V. Akinina, A. Y. Klochkov, M. B. Nikiforov, and A. V. Sokolova, "FUZZY C-MEANS ALGORITHM FOR SEGMENTATION OF AERIAL PHOTOGRAPHY DATA OBTAINED USING UNMANNED AERIAL VEHICLE," *Int. Arch. Photogramm. Remote Sens. Spat. Inf. Sci.*, vol. XL-5-W6, pp. 113–115, May 2015, doi: 10.5194/isprsarchives-XL-5-W6-113-2015.
[448] S. Cao and S. Wang, "Density-Based Fuzzy C-Means Multi-Center Re-Clustering Radar Signal Sorting Algorithm," in *2018 Eighth International Conference on Instrumentation & Measurement, Computer, Communication and Control (IMCCC)*, Jul. 2018, pp. 1843–1848. doi: 10.1109/IMCCC.2018.00380.
[449] J. P. Snyder, "Map projections: A working manual," U.S. Government Printing Office, 1395, 1987. doi: 10.3133/pp1395.
[450] J. Sun, *Python interface for OpenSky historical database*. (Sep. 01, 2022). Python. Accessed: Sep. 13, 2022. [Online]. Available: https://github.com/junzis/pyopensky
[451] The OpenSky Network, "Trino - Historical Data," The OpenSky Network API. Accessed: Nov. 11, 2026. [Online]. Available: https://openskynetwork.github.io/opensky-api/trino.html
[452] The OpenSky Network, "OpenSky Network Coverage," OpenSky Network. Accessed: Nov. 11, 2026. [Online]. Available: https://opensky-network.org/about/network
[453] B. L. Decker, "World Geodetic System 1984," Apr. 1986. Accessed: Sep. 12, 2025. [Online]. Available: https://apps.dtic.mil/sti/html/tr/ADA167570/
[454] J. Dombi and A. Dineva, "Adaptive Savitzky-Golay filtering and its applications," *Int J Adv Intell Paradig.*, vol. 16, pp. 145–156, 2020, doi: 10.1504/ijaip.2020.10027875.
[455] W. Durham, P. Belobaba, J. Cooper, and A. Seabridge, *Aircraft Flight Dynamics and Control*, 1st edition. Newark: Wiley, 2013.
[456] Federal Aviation Administration, *Helicopter Flying Handbook*, vol. FAA-H-8083-21B. 2019.
[457] "National Centers for Environmental Information (NCEI)." Accessed: Sep. 16, 2025. [Online]. Available: https://www.ncei.noaa.gov/
[458] N. G. Team, "Radar Data," National Centers for Environmental Information (NCEI). Accessed: Oct. 09, 2025. [Online]. Available: https://www.ncei.noaa.gov/maps/
[459] Amazon Web Services, Inc., "2022/01/02/KOKX/KOKX20220102_020336_V06." Accessed: Jan. 05, 2024. [Online]. Available: https://us-east-1.console.aws.amazon.com/console/home?region=us-east-1#
[460] "What is Amazon S3? - Amazon Simple Storage Service." Accessed: Sep. 16, 2025. [Online]. Available: https://docs.aws.amazon.com/AmazonS3/latest/userguide/Welcome.html#CoreConcepts
[461] "Boto3 1.40.31 documentation." Accessed: Sep. 16, 2025. [Online]. Available: https://boto3.amazonaws.com/v1/documentation/api/latest/index.html
[462] J. J. Helmus and S. M. Collis, "The Python ARM Radar Toolkit (Py-ART), a Library for Working with Weather Radar Data in the Python Programming Language," *J. Open Res. Softw.*, vol. 4, no. 1, Jul. 2016, doi: 10.5334/jors.119.
[463] R. O. C. NOAA National Weather Service, "NOAA Next Generation Radar (NEXRAD) Level II Base Data." NOAA National Centers for Environmental Information, 1991. doi: 10.7289/V5W9574V.
[464] Federal Aviation Administration, "Aircraft Registration Releasable Aircraft Database Download." 2021. [Online]. Available:

https://www.faa.gov/licenses_certificates/aircraft_certification/aircraft_registry/releasable_aircraft_download/
[465] X. Liang *et al.*, "SURF: Understanding and Predicting Urban Convection and Haze," Jul. 2018, doi: 10.1175/BAMS-D-16-0178.1.
[466] N. Dotzek, J. Gourley, and T. Fehr, *Vertical Profiles of Radar Reflectivity and Rain Rate: Observational and Modeling Results*. 2002, p. 249.
[467] P. G. Reich, "Analysis of long-range air traffic systems: separation standards—I," *J. Navig.*, vol. 19, no. 1, pp. 88–98, 1966.
[468] D. Hsu, "The evaluation of aircraft collision probabilities at intersecting air routes," *J. Navig.*, vol. 34, no. 1, pp. 78–102, 1981.
[469] D. Anderson and X. Lin, "A collision risk model for a crossing track separation methodology," *J. Navig.*, vol. 49, no. 3, pp. 337–349, 1996.
[470] International Civil Aviation Organization (ICAO), "Manual on Airspace Planning Methodology for the Determination of Separation Minima," International Civil Aviation Organization, 9689, 1998.
[471] P. G. Reich, "Analysis of long-range air traffic systems: separation standards—II," *J. Navig.*, vol. 19, no. 2, pp. 169–186, Apr. 1966, doi: 10.1017/S0373463300047196.
[472] B. Figuet, R. Monstein, M. Waltert, and J. Morio, "Data-driven mid-air collision risk modelling using extreme-value theory," *Aerosp. Sci. Technol.*, vol. 142, p. 108646, 2023.
[473] M. J. Kochenderfer, M. W. M. Edwards, L. P. Espindle, J. K. Kuchar, and J. D. Griffith, "Airspace Encounter Models for Estimating Collision Risk," *J. Guid. Control Dyn.*, vol. 33, no. 2, pp. 487–499, 2010, doi: 10.2514/1.44867.
[474] M. J. Kochenderfer, L. P. Espindle, J. K. Kuchar, and J. D. Griffith, "A comprehensive aircraft encounter model of the national airspace system," *Linc. Lab. J.*, vol. 17, no. 2, pp. 41–53, 2008.
[475] A. Weinert, "Method to Characterize Potential UAS Encounters Using Open Source Data," *Aerospace*, vol. 7, no. 11, p. 158, 2020.
[476] S. Bouarfa, H. A. Blom, R. Curran, and M. H. Everdij, "Agent-based modeling and simulation of emergent behavior in air transportation," *Complex Adapt. Syst. Model.*, vol. 1, pp. 1–26, 2013.
[477] A. Pilko, M. Ferraro, and J. Scanlan, "Quantifying Specific Operation Airborne Collision Risk through Monte Carlo Simulation," *Aerospace*, vol. 10, no. 7, p. 593, 2023.
[478] L. Barr *et al.*, "Preliminary Risk Assessment for Small Unmanned Aircraft Systems," Jun. 2017. doi: 10.2514/6.2017-3272.
[479] F. J. S. Nieto, *Collision Risk Model for High-Density Airspaces*. IntechOpen, 2020. doi: 10.5772/intechopen.89753.
[480] C. A. Roseman and B. M. Argrow, "Weather hazard risk quantification for sUAS safety risk management," *J. Atmos. Ocean. Technol.*, vol. 37, no. 7, pp. 1251–1268, 2020.
[481] B. Zhang, L. Tang, and M. Roemer, "Probabilistic Planning and Risk Evaluation Based on Ensemble Weather Forecasting," *IEEE Trans. Autom. Sci. Eng.*, vol. 15, pp. 556–566, 2018, doi: 10.1109/TASE.2017.2648743.
[482] S. Parmar and R. P. Thomas, "Effects of Probabilistic Risk Situation Awareness Tool (RSAT) on Aeronautical Weather-Hazard Decision Making," *Front. Psychol.*, vol. 11, 2020, doi: 10.3389/fpsyg.2020.566780.
[483] J. Sun, H. Vû, J. Ellerbroek, and J. Hoekstra, "Weather field reconstruction using aircraft surveillance data and a novel meteo-particle model," *PLoS ONE*, vol. 13, 2018, doi: 10.1371/journal.pone.0205029.
[484] R. Osinski and F. Bouttier, "Short-range probabilistic forecasting of convective risks for aviation based on a lagged-average-forecast ensemble approach," *Meteorol. Appl.*, vol. 25, no. 1, pp. 105–118, 2018, doi: 10.1002/met.1674.
[485] Y. Lu, M. Song, and Z. Guo, "Application and development of weather radar predicting windshear function in civil aviation," vol. 12050, pp. 120505–120505, 2021, doi: 10.1117/12.2613781.

[486] E. S. Boyarenko, E. A. Bolelov, O. V. Vasiliev, and S. S. Korotkov, “Experimental statistical analysis of radar signals reflected from weather hazards,” *Civ. Aviat. High Technol.*, 2023, doi: 10.26467/2079-0619-2023-26-5-19-29.
[487] C. Papachristos and P. J. Papakanellos, “A brief overview of a simple and efficient windshear Doppler-radar detector for aircraft early warning,” *Int J Comput Eng Res*, vol. 13, no. 1, 2023.
[488] E. Solazzo *et al.*, “Understanding Severe Weather Events at Airport Spatial Scale,” *IGARSS 2020 - 2020 IEEE Int. Geosci. Remote Sens. Symp.*, pp. 5372–5375, 2020, doi: 10.1109/IGARSS39084.2020.9323598.
[489] S. M. Ross, *A first course in probability*, vol. 2. Macmillan New York, 1976. Accessed: Sep. 30, 2024. [Online]. Available: https://toc.library.ethz.ch/objects/pdf03/UFBI_978-1-292-26920-7_01.pdf
[490] K. Sahr, D. White, and J. A. Kimerling, “Geodesic Discrete Global Grid Systems,” *Cartogr. Geogr. Inf. Sci.*, vol. 30, pp. 121–134, 2003, doi: 10.1559/152304003100011090.
[491] X. Liang, J. Ben, R. Wang, Q. Liang, X. Huang, and J. Ding, “Construction of rhombic triacontahedron discrete global grid systems,” *Int. J. Digit. Earth*, vol. 15, pp. 1760–1783, 2022, doi: 10.1080/17538947.2022.2130459.
[492] J. Zhou, J. Ben, R. Wang, M. Zheng, X. Yao, and L. Du, “A novel method of determining the optimal polyhedral orientation for discrete global grid systems applicable to regional-scale areas of interest,” *Int. J. Digit. Earth*, vol. 13, pp. 1553–1569, 2020, doi: 10.1080/17538947.2020.1748127.
[493] L. Zhao, G. Li, X. Yao, Q. Cao, and Y. Ma, “Comparison and Analysis of Hexagonal Discrete Global Grid Coding,” in *Spatial Data and Intelligence*, G. Pan, H. Lin, X. Meng, Y. Gao, Y. Li, Q. Guan, and Z. Ding, Eds., Cham: Springer International Publishing, 2021, pp. 127–133. doi: 10.1007/978-3-030-85462-1_11.
[494] X. Tong, J. Ben, Y. Wang, Y. Zhang, and T. Pei, “Efficient encoding and spatial operation scheme for aperture 4 hexagonal discrete global grid system,” *Int. J. Geogr. Inf. Sci.*, vol. 27, no. 5, pp. 898–921, May 2013, doi: 10.1080/13658816.2012.725474.
[495] H. Samet, “The Quadtree and Related Hierarchical Data Structures,” *ACM Comput Surv*, vol. 16, no. 2, pp. 187–260, Jun. 1984, doi: 10.1145/356924.356930.
[496] B. Nickerson and F. Gao, “Spatial Indexing of Large Volume Swath Data Sets,” *Int J Geogr Inf Sci*, vol. 12, pp. 537–559, 1998, doi: 10.1080/136588198241662.
[497] T. Kim, J. Lee, K.-S. Kim, A. Matono, and K.-J. Li, “Utilizing extended geocodes for handling massive three-dimensional point cloud data,” *World Wide Web*, pp. 1–24, 2020, doi: 10.1007/s11280-020-00783-1.
[498] K. Karydis, I. Poulakakis, J. Sun, and H. G. Tanner, “Probabilistically valid stochastic extensions of deterministic models for systems with uncertainty,” *Int. J. Robot. Res.*, vol. 34, no. 10, pp. 1278–1295, 2015.
[499] International Civil Aviation Organization (ICAO), “Performance Based Navigation (PBN) Manual (Doc 9613),” 9613, 2013. Accessed: Jul. 20, 2022. [Online]. Available: https://store.icao.int/en/performance-based-navigation-pbn-manual-doc-9613
[500] J. Rohmer, “Uncertainties in conditional probability tables of discrete Bayesian Belief Networks: A comprehensive review,” *Eng. Appl. Artif. Intell.*, vol. 88, p. 103384, 2020.
[501] D. P. Kroese, T. Brereton, T. Taimre, and Z. I. Botev, “Why the Monte Carlo method is so important today,” *Wiley Interdiscip. Rev. Comput. Stat.*, vol. 6, no. 6, pp. 386–392, 2014.

Appendix A – DGGS framework theoretical background used to support the methodology

This appendix presents the theoretical background supporting the DGGS-based 4D airspace model construction described in Chapter 4. It summarizes the principal design considerations underlying the selected grid geometry, coordinate projection, subdivision, data mapping, and spatial indexing methods. Figure 106 illustrates the main steps involved in establishing the DGGS framework.

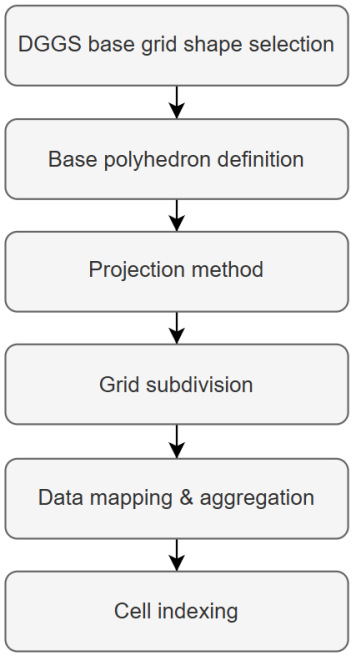


Figure 106: Main steps for creating the DGGS framework

A.1 DGGS base grid shape selection

The *base grid shape* determines the form of the grid cells and directly influences how spatial data is represented and analyzed. DGGS can be implemented using different types of base grid shapes, each with unique properties suited to specific applications. Table 28 summarizes the main advantages, disadvantages, and most common applications for each DGGS base grid shape.

Table 28: Comparison of DGGS basic shapes and their properties [490], [491]

DGGS basic shape	Advantages	Disadvantages	Main applications
Square	• **Ease of computation**: Square grids align naturally with Cartesian (UTM) coordinate systems, which simplifies operations like snapping data points to grids and performing grid-based calculations. • **Efficient data representation**: Suitable for spatial indexing methods like Morton (Z-order) codes, which are simpler to implement for square grids. • **Adaptability**: Square grids allow straightforward multi-resolution analysis (quadtree decomposition) for managing data at different spatial scales. • **Compatibility**: Compatible with existing GIS tools and algorithms that are built around rectangular raster grids.	• **Non-uniform neighbor distances**: Adjacent cells have varying distances depending on direction (diagonal vs. orthogonal), which can complicate adjacency-based computations. • **Distortion**: Increased distortion at higher latitudes when used for global applications due to the rectangular shape not conforming to spherical surfaces. • **Edge effects**: Gaps and overlaps can occur when mapping square grids onto a curved surface without additional corrections.	• **Raster-based analyses**: Processing satellite imagery, land cover classifications, and elevation data due to alignment with raster data formats. • **Urban and regional planning**: Square grids are used in scenarios where local projections like UTM are prevalent, making them ideal for air traffic management (ADS-B data) and weather modelling. • **Environmental modelling**: Flood mapping or deforestation studies often use square grids for straightforward overlay and spatial analysis.
Hexagonal	• **Uniform neighbor relationships**: Each hexagon has six equidistant neighbors, making spatial computations like proximity searches more consistent. • **Equal area**: Reduces sampling bias and edge effects, which is crucial for modelling continuous phenomena. • **Compactness**: The perimeter-to-area ratio is lower than squares, making hexagons efficient for area-based modelling.	• **Complexity**: More difficult to implement and integrate with existing systems that predominantly use Cartesian coordinates. • **Storage inefficiency**: Hexagonal grids require more sophisticated indexing systems, such as custom Morton codes or hex-tree structures. • **Projection distortion**: Mapping a hexagonal grid onto a spherical surface introduces challenges in ensuring equal area across all cells.	• **Meteorology**: Modelling weather patterns or radar data because of uniform sampling and smoother transitions. • **Ecology and resource management**: Habitat modelling or resource allocation, where equal-area representation is critical. • **Hydrology**: River basin modelling due to the natural flow patterns supported by hexagons.
Triangular	• **Alignment with spheres**: Triangles are easier to map onto curved surfaces (spheres) with minimal distortion, making them ideal for global applications. • **Precision**: Triangular grids support finer resolutions through recursive subdivision. • **Adaptability**: Efficient for handling data that spans across diverse geographic extents (from equator to poles).	• **Irregular neighbor relationships**: Unlike hexagonal grids, triangles have inconsistent distances and angles between neighbors, which can complicate computations. • **Higher computation costs**: Operations like interpolation and adjacency determination are more computationally intensive than squares or hexagons. • **Complex indexing**: Requires specialized hierarchical encoding, such as triangular quadtrees.	• **Global-scale applications**: Earth observation or satellite data analysis, where minimizing distortion across the globe is a priority. • **Astronomy and cosmology**: Mapping celestial data onto a spherical coordinate system. • **Global Navigation Systems**: Handling geospatial data that spans large geographic extents.

Rhombus (diamond)	• **Balance between distortion and simplicity**: Rhombus grids can provide uniformity over certain types of datasets. • **Simplified hierarchical refinement**: Rhombus grids support quadtree-like operations for efficient data representation.	• **Limited use cases**: Not as widely supported or implemented in existing GIS tools. • **Projection challenges**: Mapping rhombus grids onto spheres can create uneven distortion without careful projection adjustments.	• **Remote sensing**: Large-scale raster data processing where minimizing edge effects is important. • **Multi-resolution analysis**: Use cases requiring adaptable resolution for zooming or panning in visualizations. • **Specialized applications**: Certain ecological or resource management studies benefit from the balance between distortion and compatibility.
Mixed	• **Customizability**: Mixed grids combine the strengths of different shapes, allowing for tailored solutions to specific challenges (balancing distortion and computation efficiency). • **Localized optimization**: Specific regions (polar areas) can use a more suitable shape while maintaining a consistent global structure. • **Flexibility**: Can be designed to minimize distortion and maximize computational efficiency across diverse datasets.	• **Implementation complexity**: Combining multiple grid types requires sophisticated algorithms and data management strategies. • **Inconsistencies in operations**: Spatial operations can become complicated due to differences in neighbor relationships or cell sizes between grid types. • **Projection-specific design**: Often tied to specific projections or datasets, limiting general applicability.	• **Specialized/Custom geospatial applications**: Designed for specific problems where no single grid type suffices (modelling ocean currents and coastal interactions simultaneously). • **Polar studies**: Mixing hexagons and triangles to handle polar regions without introducing excessive distortion. • **Global environmental monitoring**: Applications that require a combination of precision, uniformity, and adaptability.

The proposed methodology uses a square grid shape due to its compatibility with rectangular raster grids (i.e., UTM and Cartesian coordinates) and its computational simplicity. Additionally, the airspace model is not intended to be used for global analysis of the Earth but rather for localized airspace regions of varying resolutions spanning a single UTM zone. Extension to regions spanning multiple UTM zones would require additional coordinate harmonization or a suitable common spatial reference. Square grids provide straightforward indexing and alignment with geospatial data already projected in the UTM system. Unlike hexagonal grids, square grids integrate seamlessly into existing geospatial workflows without requiring additional mapping to new coordinate systems. This approach provides a practical planar representation with limited projection distortion over the localized regions considered in this thesis.

A.2 Base polyhedron definition

The *base polyhedron* in a Discrete Global Grid System is the 3D shape onto which the base grid shape from the previous step is applied or from which it is derived [492]. It serves as the geometric foundation upon which the Earth's surface is mapped and subdivided into grid cells and is a 3D shape (polyhedron) chosen to approximate the spherical shape of the Earth while providing a framework for creating a structured grid system.

The choice of a base polyhedron depends on its compatibility with the selected cell geometry and the geometric distortion introduced when representing the Earth's surface. Cubic polyhedra are naturally compatible with quadrilateral subdivisions because their faces can be partitioned into regular square cells [492]. However, the implementation developed in this thesis uses a localized Cartesian representation based on UTM coordinates rather than explicitly projecting the Earth's surface onto cube faces. The resulting

three-dimensional airspace grid consists of rectangular volumetric cells whose horizontal dimensions are defined in projected coordinates and whose vertical dimension represents altitude.

For local/regional applications, a simpler base polyhedron (cube for square grids) is often sufficient because distortion concerns are minimal [492]. On the other hand, for global applications, a more complex base polyhedron that better approximates a sphere (icosahedron for triangular or hexagonal grids) is preferred to ensure consistent coverage and reduce distortion across the entire sphere [492].



Figure 1. Platonic solids and their equal-area sphere: the tetrahedron, cube, octahedron, dodecahedron, and icosahedron.

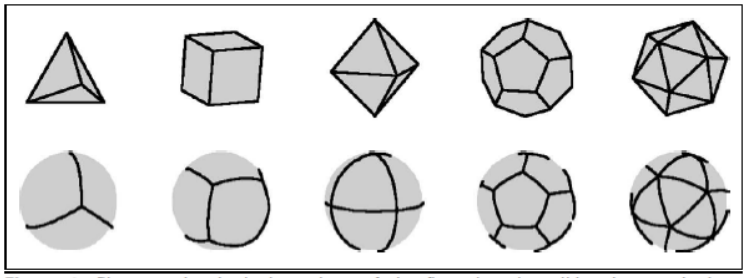


Figure 2. Planar and spherical versions of the five platonic solids: the tetrahedron, hexahedron (cube), octahedron, dodecahedron, and icosahedron.

Figure 107: Comparison of different base polyhedrons (taken from [490], [491])

A.3 Projection method

The *projection method* maps the Earth's spherical surface onto the flat faces of the base polyhedron. For a cube, each face represents a planar surface onto which geospatial coordinates are projected.

In the developed methodology, the UTM coordinate system serves as the projection base, bypassing the need for traditional DGGS projection methods like Snyder equal-area projection [493].

UTM uses a transverse Mercator projection, which directly maps geographic coordinates (ϕ, λ) (latitude, longitude) to Cartesian coordinates (x, y) with minimal distortion within a zone. If the airspace model spans only one UTM zone, the planar coordinates are calculated as:

$$x = (E - E_0), \qquad y = N \tag{21}$$

where E and N are the easting and northing coordinates in meters, and E_0 is the central meridian offset for the specific UTM zone. A more generalized calculation of the planar coordinates can be obtained which can span multiple UTM zones using:

$$x = k_0 N(\lambda - \lambda_0)cos\phi \tag{22}$$

$$y = k_0 N ln(\tan(\frac{\pi}{4} + \frac{\phi}{2})) \tag{23}$$

Where k_0 is the scale factor, N is the radius of curvature of the ellipsoid, and λ_0 is the central meridian of the UTM zone. This projection approach minimizes distortions and ensures alignment of grid cells with existing UTM zone boundaries.

A.4 Grid subdivision

An important feature of a DGGS framework is its use of a *hierarchical grid system* that enables representations of data at multiple levels of detail, making it possible to zoom in or out on spatial datasets using a single airspace model. Making use of hierarchical grid systems also organizes data spatially in a way that optimizes storage and retrieval by narrowing the search space [494]. Hierarchical grids can also adapt to varying data densities, providing finer grids where data is dense and coarser grids where data is sparse. Using a hierarchical grid also simplifies parts of the development of visualization tools for the airspace model by structuring that data into levels that match user needs (variable granularity based on dataset density and desired results for the case study).

Different methods exist to create hierarchical grid systems. Hierarchical grid subdivision can be implemented through quadtree or octree structures. A quadtree recursively divides a two-dimensional cell into four children, whereas an octree subdivides a three-dimensional cell into eight volumetric children [495]. The proposed airspace representation uses a three-dimensional rectangular grid, for which octree subdivision provides the corresponding hierarchical interpretation. This hierarchical 3D structure efficiently supports spatial indexing, reduces memory requirements, and simplifies data representation and multiscale visualization for volumetric and geospatial datasets. This method was selected for its compatibility with the cubic base polyhedron shape. The Quadtree subdivision method can be summarized by the following 3 steps:

1. The process begins with the largest cell size (100 km cells) and progresses to finer resolutions (1 km or smaller) based on the spatial resolution needs of the airspace model. Divide each face of the cube into n×n squares (n=2 for initial resolution).
2. Recursively subdivide each square. At each resolution level k, the number of cells increases as:

$$n_{level} = 4^k \tag{24}$$

3. Assign hierarchy levels. Each cell is labelled with its position in the hierarchy (face_id, row, column)

This hierarchical approach ensures adaptability across varying spatial scales while maintaining efficient data management.

A.5 Data mapping and aggregation

Data mapping is the process of allocating a DGGS cell (cell ID) to each geospatial data point in a dataset and aggregating the data from multiple data points within a single cell. Each processed airsapec entity datapoint is associated with a DGGS cell using the cell-assignment formulation presented in Equation (25). For UTM coordinates, the cell assignment is determined by:

$$CellID = \left(\left\lfloor \frac{x}{\Delta x} \right\rfloor, \left\lfloor \frac{y}{\Delta y} \right\rfloor\right) \tag{25}$$

Where x and y are the UTM coordinates, Δx and Δy are the cell dimensions in meters, and $\lfloor _ \rfloor$ is the floor function.

Once data points are mapped to cells, the next task is *aggregation*, where data within each cell is aggregated. This step simplifies analysis by reducing the raw data to manageable, representative values. Different aggregation methods are used based on the case study and type of the selected geospatial entity.

A.6 Cell indexing

Spatial indexing (or cell indexing) is a computational method used to organize and manage spatial data efficiently, enabling quick retrieval and analysis of geospatial data. A *spatial index* is a data structure that improves the performance of spatial queries, such as finding objects within a certain area (range queries), identifying the nearest neighbors to a point, or performing intersections between shapes. Spatial indexing helps convert multi-dimensional spatial problems into more computationally efficient one-dimensional problems. Encoding the spatial data allows efficient spatial queries and retrievals within the DGGS.

Several spatial indexing methods are available for hierarchical and multidimensional grids [496]. The proposed implementation uses *Morton (Z-order) indexing* [497], which combines discrete spatial cell coordinates into a one-dimensional identifier through bit interleaving. For the three-dimensional airspace representation, the coordinate indices corresponding to the x, y, and z directions are interleaved to form the Morton code. This approach preserves a degree of spatial locality in the resulting ordering, facilitating the organization and processing of nearby cells.

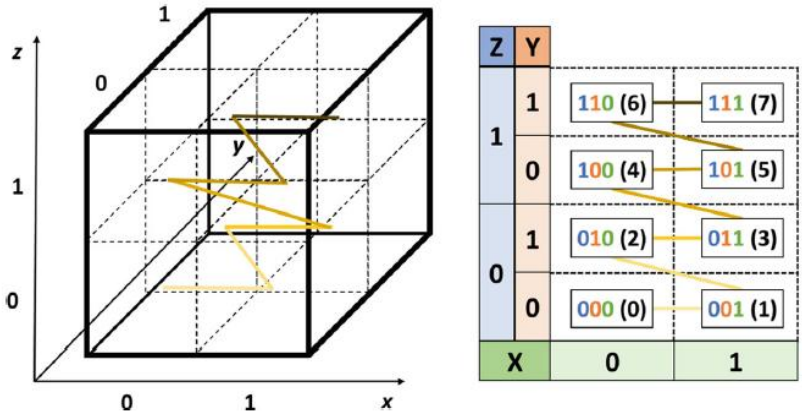


Figure 108: Visual representation of 1-resolution level Morton codes in a 3D space for a cubic grid (taken from [497])

The Morton encoding process can be described as:

1. Convert the x and y UTM coordinates to binary:

$$x = binary(E), \qquad y = binary(N) \tag{26}$$

For example, if x = 2 and y =3:

$$x = 10_2, \qquad y = 11_2 \tag{27}$$

2. Interleave the binary digits of x and y to form the Morton code:

$$Morton(x, y) = interleave(x, y) \tag{28}$$

For example, if $x = 10_{10} = 1010_2$ and $y = 12_{10} = 1100_2$, the resulting Morton code is:

$$Morton(x, y) = 11001100 = 111000_2 = 56_{10} \quad (29)$$

3. Assign the Morton code as the unique identifier for the cell.

This encoding allows efficient spatial queries and retrievals within the DGGS.

Appendix B – Supporting literature review for location-based airspace risk analysis

B.1 Purpose and scope

Chapter 2 reviewed the literature on airspace models broadly. This appendix focuses more narrowly on risk analysis models and risk metric formulations that support Experiment #2, as discussed in Chapter 6. Its purpose is to justify the development of the location-based risk metrics used in Chapter 6 without overloading the main experiment chapter with a long risk model literature review.

In this thesis, *location-based airspace risk* refers to a time-varying spatial field in which locations in a 4D airspace model can be assigned risk metric values. The source of risk may be mid-air collision (MAC) risk, hazardous weather, or a combination of multiple hazards. The risk metrics developed in Chapter 6 are therefore intended to test whether the DGGS-based airspace model can produce interpretable risk fields from historical aircraft and weather data.

The appendix focuses on two major risk model families. The first concerns MAC risk. The second concerns weather-related operational risk. These two families are reviewed because they provide the basis for the two single-hazard risk metrics used in Chapter 6 and for the combined risk metric developed from them.

B.2 MAC risk models

MAC risk models in aviation have evolved from early deterministic approaches [467], [468], [469] to more complex probabilistic models such as the Collision Risk Model (CRM) used by the International Civil Aviation Organization (ICAO) [470]. Deterministic models provide specific outcomes based on defined inputs and scenarios, assuming that conditions remain constant or predictable [471]. Because they do not explicitly represent randomness or variability in real-world scenarios [498], they are often best suited to well-defined operating contexts.

The Reich model and its derivatives [467], [468], [469] are examples of deterministic MAC models that have been used as a basis for defining separation minima [413] and route spacing [499]. These models provide important analytical foundations for collision risk analysis. However, they are typically tied to specific operating environments, route structures, surveillance assumptions, and separation standards. This limits their direct reuse for the purpose of Experiment #2, where risk must be computed at fixed DGGS centroids in a local 3D airspace model rather than along predefined routes or within a specific separation-standard framework.

A second group of MAC risk models uses probabilistic formulations. These models incorporate randomness and uncertainty and evaluate a range of possible outcomes and their probabilities [472]. Probabilistic approaches are useful because they can represent uncertainty in aircraft behaviour, environmental conditions, surveillance quality, and future aircraft states. Some probabilistic MAC risk models are also combined with Monte Carlo simulation [473], [474], [475], agent-based modelling [31, 32], or Bayesian networks [33, 34]. These approaches can provide richer representations of uncertainty than deterministic models.

However, probabilistic MAC models can also require substantial scenario-specific assumptions. In some cases, expert judgement is needed to define probability distributions, encounter assumptions, or operational behaviours [500]. Monte Carlo-based models can also require a large number of simulation trials to estimate

rare events with stable results [501]. This can make them computationally expensive when the analysis must scale across many locations, time steps, or aircraft interactions.

For Experiment #2, the main takeaway is that the MAC risk literature provides important foundations for aircraft interaction analysis, but many existing MAC models are not designed to produce a reusable DGGS-based 3D risk field at fixed airspace locations and time steps. The MAC metric in Chapter 6 is therefore not intended to replace these models. It is one location-based implementation that uses observed aircraft positions and convergence behaviour to identify where aircraft interactions create elevated MAC-related risk at DGGS centroids.

B.3 Weather-related operational risk models

The second major risk model family concerns weather-related operational risk. Weather risk models use weather observations, forecasts, radar products, ensemble predictions, or mission-specific thresholds to evaluate how hazardous weather affects aviation operations. These models are important because hazardous weather can reduce airspace capacity, disrupt trajectories, increase controller workload, and create safety-relevant operating conditions.

The reviewed literature includes several types of weather risk models. Fine-scale sUAS weather risk models quantify weather hazards using high-resolution forecasts and flight-data constraints [480]. Ensemble weather forecasting frameworks use probabilistic analysis for mission planning and risk evaluation [481]. NEXRAD-based tools such as the Risk Situation Awareness Tool (RSAT) use radar imagery to improve weather-related risk awareness [482]. Other approaches reconstruct or estimate local weather fields using aircraft surveillance data, which demonstrates the value of time-varying weather analysis in operational airspace contexts [483].

Additional weather risk models focus on short-range convective forecasts [484], windshear prediction [485], radar-based hazard classification [486], [487], and severe-weather analysis near airports [488]. These models show that weather risk is spatially distributed, time-varying, and often three-dimensional. This supports the use of radar-derived storm cell data in Experiment #2.

However, weather risk models are often developed separately from MAC risk models. They frequently use different assumptions, data structures, spatial representations, and output formats. Many weather risk models are designed for forecasting, decision support, or operational impact assessment, rather than for producing a combined aircraft-weather risk field on a common DGGS grid.

For Experiment #2, the main takeaway is that weather risk models provide strong justification for using radar-derived storm cell data, but they do not directly provide the combined DGGS-based risk field required by this thesis. Chapter 6 therefore uses storm cell motion relative to DGGS centroids as one implementation of a location-based weather risk metric.

B.4 Requirements identified for the Experiment #2 risk model

Six risk model requirements were identified from the reviewed MAC and weather risk literature. These requirements define what the Experiment #2 risk model must demonstrate. They are not used to claim that the model developed in Chapter 6 is a universally optimal risk model. Instead, they define the specific capabilities needed to test the DGGS-based airspace metric computation step of the thesis methodology.

Table 28 summarizes these requirements, their meaning in Experiment #2, and why they matter.

Table 29: Risk model requirements identified for Experiment #2

Requirement	Meaning in Experiment #2	Why it matters
Air-vehicle agnostic	The location-based MAC metric uses observed aircraft position and time data rather than aircraft-specific aerodynamic or performance models.	Allows different observed aircraft types to be evaluated within the same formulation when suitable trajectory data are available.
Applicable to different airspace regions	The metric is computed on DGGS cells and can be reapplied by changing the model boundary, resolution, time interval, and input data.	Avoids tying the risk model to one fixed route structure, airport geometry, or predefined sector design.
Fully three-dimensional	Risk is computed using horizontal position and altitude.	Prevents altitude-dependent risk from being hidden in a 2D representation.
Time-varying	Risk is computed at selected time steps.	Allows risk to increase, decrease, appear, or disappear as aircraft and weather move. Enables time-dependent risk analysis.
Scalable in space, time, and entity count	The same risk logic can be applied to different model sizes, time windows, and numbers of entities, subject to computational limits.	Supports the thesis goal of analysing large historical airspace datasets and extending the method to more complex scenarios.
Multi-hazard capability	MAC risk and weather risk can be computed on the same DGGS grid and combined.	Allows aircraft-related and weather-related risk to be interpreted within the same location-based airspace framework.

B.5 Summary and implication for Experiment #2

The reviewed literature shows that MAC risk and weather-related operational risk are often treated as separate problem classes. Existing models address important parts of the risk analysis problem, including deterministic collision risk formulation, probabilistic conflict analysis, simulation-based safety assessment, ensemble weather forecasting, radar-based weather risk awareness, and weather decision support. However, the reviewed literature did not identify a single existing model that directly provides the capability tested in Chapter 6: a DGGS-based, location-based, three-dimensional, time-varying, multi-hazard risk field computed from historical aircraft and weather data.

The risk model developed in Chapter 6 provides a specific implementation of the location-based metric computation component of the proposed methodology. Using historical aircraft and weather observations, it demonstrates the derivation of independently calculated MAC risk and weather risk indicators and their integration into a combined analytical field. The experiment thereby evaluates the capacity of the reusable DGGS-based 4D airspace model to support multiple spatially and temporally referenced analytical products within a common representation.

Appendix C – Supporting clustering algorithm comparison and KMeans selection

Appendix C documents the supporting clustering-algorithm comparison used to select KMeans for the behaviour-based aircraft categorization application evaluated in Experiment #3. The comparative methodology and results are reported in [3] and are retained here to provide methodological traceability for the implementation choice used in Chapter 7. The appendix is separated from the principal Experiment #3 results because the thesis-level objective of Experiment #3 is to evaluate whether recurring operational patterns can be identified and interpreted from airspace-model trajectory outputs, rather than to establish a generally superior clustering algorithm.

The supporting comparison evaluates DBSCAN, HDBSCAN, KMeans, and GMM using a common fixed-length trajectory representation. Candidate configurations are evaluated using complementary internal cluster-validity metrics, trajectory coverage, effective cluster count, and qualitative interpretation of the resulting trajectory structure. The combined evidence is then used to select the clustering implementation applied in the principal Chapter 7 case study.

The comparative study reported in [3] uses a final sample of 343 trajectory units, whereas the finalized categorization analysis presented in Chapter 7 uses 346 trajectory units. The numerical results reproduced in this appendix are therefore retained as reported in the supporting study, while the principal Experiment #3 results and validation use the dataset described in Section 7.2.

C.1 Dataset characterization

The supporting algorithm comparison uses the same general five-hour LGA trajectory-analysis application and fixed-length trajectory representation described in Section 7.2.2. The aligned trajectories exhibit substantial spatial and vertical heterogeneity, including dense low-altitude terminal movements and less frequent higher-altitude trajectories. This combination provides a useful test case because different clustering algorithms must simultaneously represent dominant high-density trajectory structure and lower-frequency movement patterns.

The histograms in Figure 109 quantify the non-uniform spatial and altitude distributions of the supporting dataset. Horizontal X and Y observations are concentrated within recurrent terminal-area corridors, while the altitude distribution is dominated by low-altitude terminal traffic with a substantially smaller higher-altitude tail. Figure 110 provides the corresponding boxplot representation and further illustrates this imbalance. Points outside the boxplot whiskers represent statistical outliers under the boxplot convention and should not be interpreted as erroneous observations because trajectory-quality processing has already been applied.

Exploratory inspection suggested several broad recurring movement families within the supporting dataset. These visually identified families are used only as a loose interpretive reference when identifying severe under-segmentation or excessive fragmentation. They are neither supplied to the clustering algorithms as labels nor treated as ground-truth categories.

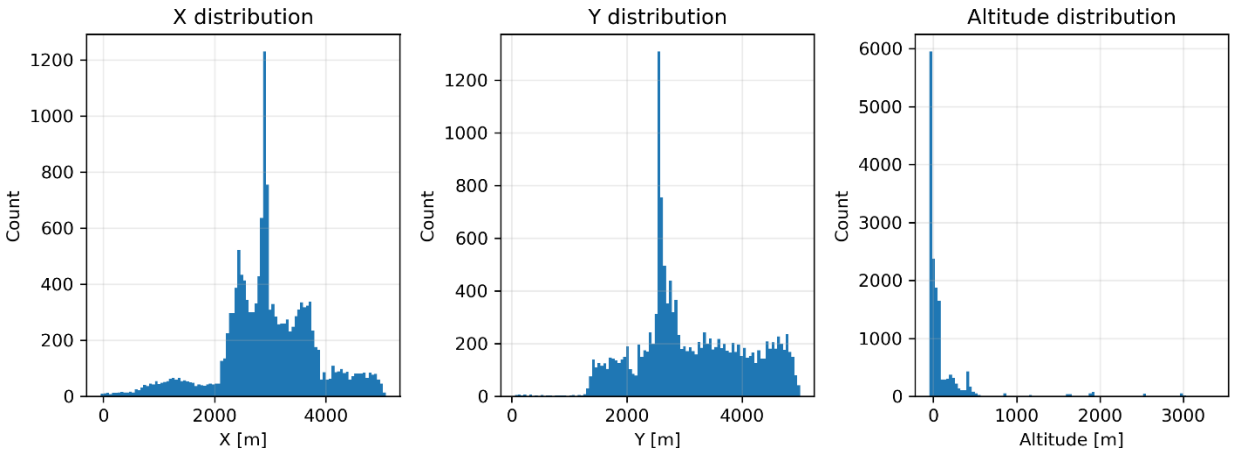


Figure 109: Histograms of the x, y and z(altitude) data distributions

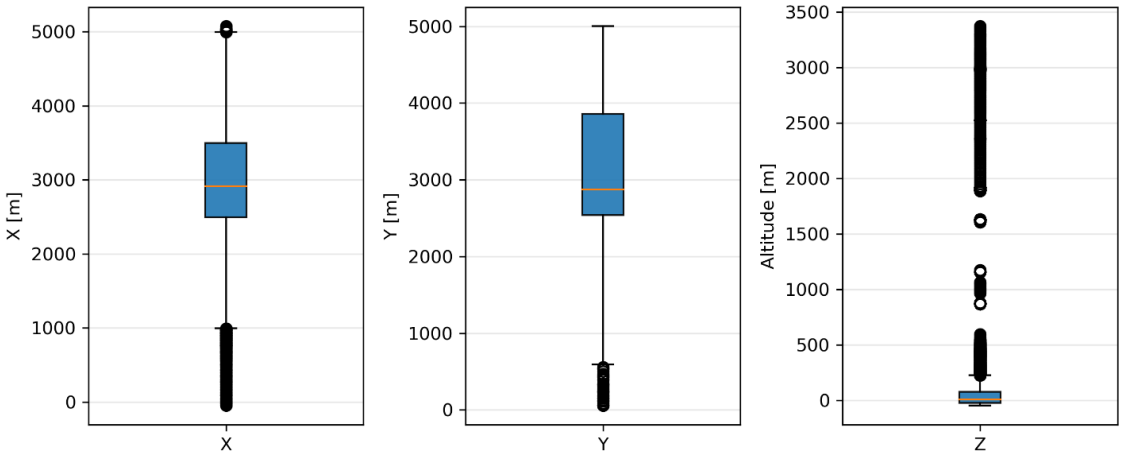


Figure 110: Boxplots of the x, y and z(altitude) data distributions

C.2 Candidate clustering algorithm comparison

The supporting comparison evaluates four clustering algorithms—DBSCAN, HDBSCAN, KMeans, and GMM—to determine which tested approach provides the most appropriate implementation for the behaviour-based aircraft categorization application. The algorithms represent different clustering assumptions: density-based, hierarchical density-based, partitioning-based, and probabilistic/model-based approaches, respectively. The objective is to evaluate their interaction with the supporting trajectory dataset and intended analytical use rather than to identify a universally preferred clustering method.

For clarity, **clustering algorithm** refers to the algorithm family (e.g., KMeans or DBSCAN); **candidate configuration** refers to one application of an algorithm using a specific combination of hyperparameter values; **shortlisted configuration** refers to a retained candidate selected for the quantitative comparison; and **representative configuration** refers to the single configuration from each algorithm family selected for detailed visual comparison.

C.2.1 Candidate algorithms and parameter search

Four candidate clustering algorithms are evaluated: DBSCAN, HDBSCAN, KMeans, and GMM. These algorithms were selected because they represent different clustering assumptions. DBSCAN and HDBSCAN are density-based methods. KMeans is a partitioning-based method. GMM is a probabilistic model-based method. Table 30 summarizes the candidate algorithms and their role in the comparison.

The candidate search is designed to characterize how the four clustering algorithms respond to the supporting trajectory dataset under different hyperparameter settings. By examining a range of configurations for each algorithm, the comparison identifies differences in cluster separation, trajectory coverage, category granularity, and operational interpretability. These results provide the basis for selecting an implementation suited to the behaviour-based aircraft categorization application.

Table 30: Clustering algorithms tested in Experiment #3

Algorithm	Clustering family	Purpose in comparison
DBSCAN	Density-based	Test whether dense trajectory groups can be identified while treating lower-density trajectories as noise
HDBSCAN	Hierarchical density-based	Test whether density-based structure can be recovered more robustly across varying local densities
KMeans	Partitioning-based	Test whether broad trajectory behaviour families can be represented using a compact full-assignment model
GMM	Probabilistic/model-based	Test whether a probabilistic model can recover flexible trajectory groupings

The hyperparameters of each algorithm are varied during the candidate search, as summarized in Table 31. For KMeans, the number of clusters is varied to test whether a compact partitioning model can recover broad behaviour families in the trajectory dataset. For DBSCAN, the neighbourhood radius and minimum sample size are varied to test whether density-based clustering can isolate dense trajectory groups while handling lower-density trajectories. For HDBSCAN, the minimum cluster size, minimum samples, and cluster selection method are varied to test whether a hierarchical density-based method can provide a more stable alternative to DBSCAN. For GMM, the number of components and covariance structure are varied to test whether a probabilistic clustering model can identify behaviour families with more flexible cluster shapes.

Table 31: Description of the hyperparameters varied for each candidate clustering algorithm

Clustering Algorithm	Hyperparameter	Hyperparameter Description	Candidate hyperparameter test values
KMeans	k	Number of clusters.	3, 4, 5, 6, 7, 8, 9, 10
	Initial conditions	Method used to initialize cluster centers	k-means++
	Seeding	Random seed for reproducibility.	42
	n_init	Number of repeated initializations	10
DBSCAN	ε (epsilon)	Neighborhood radius used to define proximity between points.	0.5, 0.7, 0.9, 1.1, 1.3, 1.5, 1.7, 2.0, 2.5, 3.0 (for scaled data)
	minPts	Minimum number of points required to form a dense region.	5, 8, 10, 12, 15, 18, 20, 25, 30
HDBSCAN	min_cluster_size	Minimum number of points in a cluster (smaller clusters are considered noise).	3, 5, 8, 10, 15, 20, 25, 30
	min_samples	Controls how conservative the clustering is with respect to noise (larger values → more points labelled as noise).	3, 5, 8, 10, 15, 20, 25, 30
	cluster_selection_method	Method used by HDBSCAN to extract final clusters from the condensed hierarchy.	eom, leaf
Gaussian Mixture Models (GMM)	Number of components	Number of Gaussian components.	3, 4, 5, 6, 7, 8, 9, 10
	Covariance type	Covariance structure of the Gaussian components	full, diag
	Seeding	Random seed used to initialize the probabilistic model and ensure reproducible results.	42

The comparison follows a multi-stage evaluation procedure. A broad candidate pool is first generated by sweeping the hyperparameter ranges summarized in Table 31. Non-informative configurations are then removed, including solutions with fewer than two effective clusters or insufficient valid samples for calculation of the internal validity metrics. The retained configurations are subsequently evaluated to identify algorithm–dataset trends across the tested parameter ranges.

Quantitative evaluation uses the Silhouette coefficient, Davies–Bouldin index, Calinski–Harabasz index, effective cluster count, and, for density-based algorithms, noise fraction. A reduced shortlist of five configurations from each algorithm family is used to summarize the principal quantitative trends. One representative configuration from each algorithm family is then selected for detailed qualitative comparison using two-dimensional trajectories, three-dimensional trajectories, and aligned altitude profiles. Final algorithm selection considers the combined balance of quantitative separation, trajectory coverage, stability across the tested settings, computational practicality, and operational interpretability rather than the optimum value of a single internal metric.

- The **Silhouette coefficient** measures how well each trajectory fits within its assigned cluster relative to neighbouring clusters. Higher values indicate better within-cluster cohesion and between-cluster separation.
- The **Davies-Bouldin index** measures similarity between clusters. Lower values indicate more compact and more distinct clusters.
- The **Calinski-Harabasz index** compares between-cluster dispersion with within-cluster dispersion. Higher values indicate better global separation.
- The **number of effective clusters** measures how many non-empty clusters are produced by the algorithm. For DBSCAN and HDBSCAN, the noise class is excluded from this count. This metric is used to determine whether a candidate model under-segments the dataset into too few groups or over-fragments it into too many groups.
- The **noise fraction** measures the proportion of trajectories assigned to the noise class. This metric applies only to DBSCAN and HDBSCAN because these methods can label some trajectories as noise. This metric is important for the aircraft categorization application because a configuration with strong internal validity scores may still be less suitable if it leaves a substantial proportion of valid, lower-density trajectories unassigned.

C.2.2 Quantitative candidate comparison

To preserve readability while still representing the main algorithm-level trends observed during candidate screening, a reduced quantitative shortlist is used for the figure-based comparison. The shortlist contains five shortlisted configurations for each algorithm family. Quantitative evaluation metrics remain the main basis for comparison, while the shortlist also considers the number of effective clusters and, for the density-based methods, trajectory coverage through the noise fraction.

The quantitative results presented in Figures 111–115 were obtained from the supporting clustering-algorithm comparison reported in [3], using the fixed-length trajectory representation described in Section D.1. For each algorithm, candidate configurations were evaluated using the selected internal validity and trajectory-coverage metrics. The five shortlisted configurations per algorithm provide a reduced representation of the results obtained during the broader hyperparameter search. These configurations are distinct from the single representative configuration subsequently examined through trajectory visualizations in Section D.2.3.

The approximate five-family structure identified during exploratory inspection is used only as contextual evidence when examining strong under-segmentation or fragmentation. It is not used as a target value of K, a supervised label, or an independent measure of clustering accuracy. Configurations producing fewer or more groups are therefore retained when they provide useful evidence of broader or finer trajectory structure.

Figure 111 compares the Silhouette coefficients of the five shortlisted candidates from each algorithm. DBSCAN spans 0.64–0.84 and contains the highest individual Silhouette coefficient among the displayed candidates. The five HDBSCAN EOM candidates remain constant at 0.75. KMeans spans 0.58–0.70, while full-covariance GMM spans 0.58–0.71. These results show that DBSCAN can produce particularly strong cluster separation, but also exhibits greater variation across the retained parameter settings, whereas the displayed HDBSCAN EOM solutions are highly consistent. Because trajectories labelled as noise are excluded from the internal cluster-validity calculations for DBSCAN and HDBSCAN, these results must be interpreted together with the trajectory-coverage results in Figure 112.

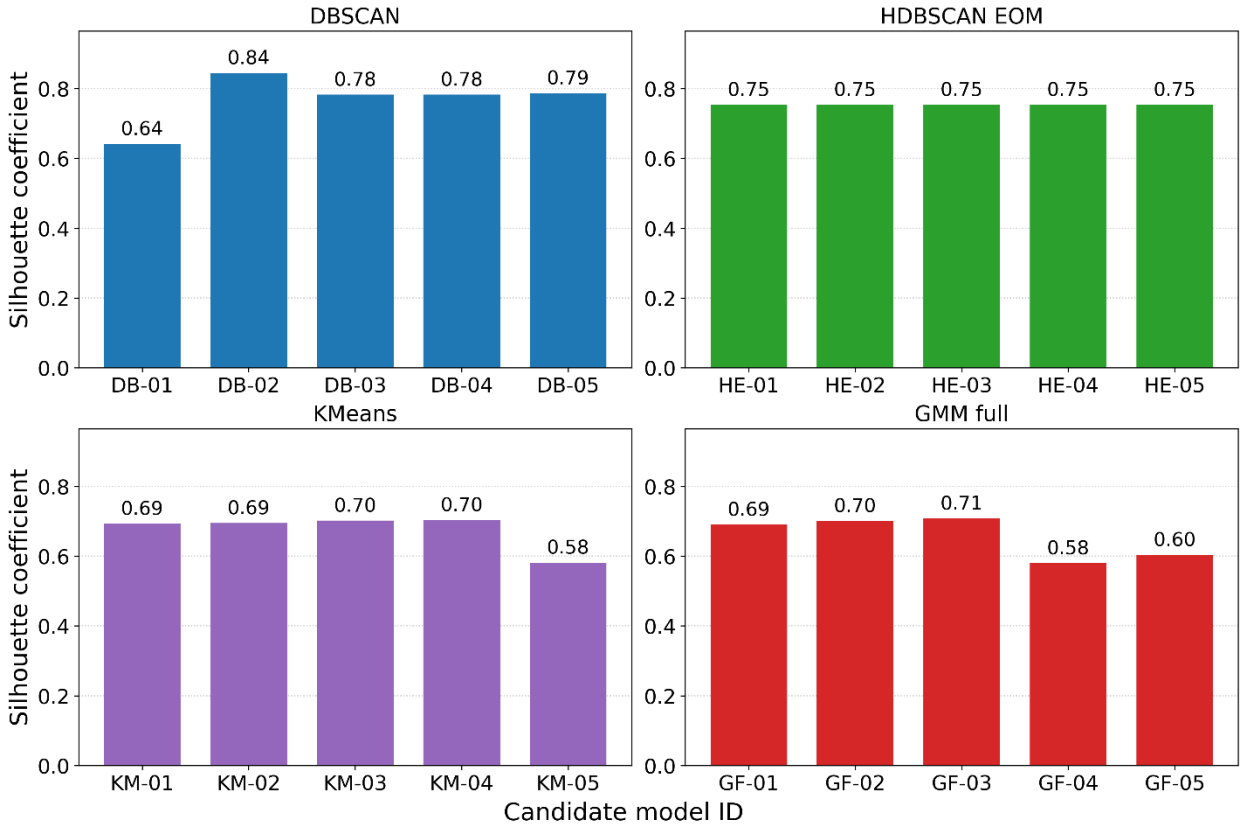


Figure 111: Comparison of Silhouette coefficient values for the 5 shortlisted candidates from each clustering algorithm

Figure 112 compares the noise fractions of the five shortlisted density-based candidates. The DBSCAN configurations classify 11.4–47.5% of the trajectories as noise, demonstrating substantial sensitivity of trajectory coverage to the selected DBSCAN parameters. In contrast, all five HDBSCAN EOM candidates classify 7.9% of the trajectories as noise. HDBSCAN therefore provides substantially more consistent trajectory coverage across the retained configurations. This distinction is important for the aircraft categorization objective because trajectories classified as noise are not assigned to a behaviour category.

The noise fractions presented in Figure 112 describe the five shortlisted configurations used for quantitative comparison. The representative configurations selected for the subsequent visual analysis are evaluated separately and may therefore exhibit different noise fractions.

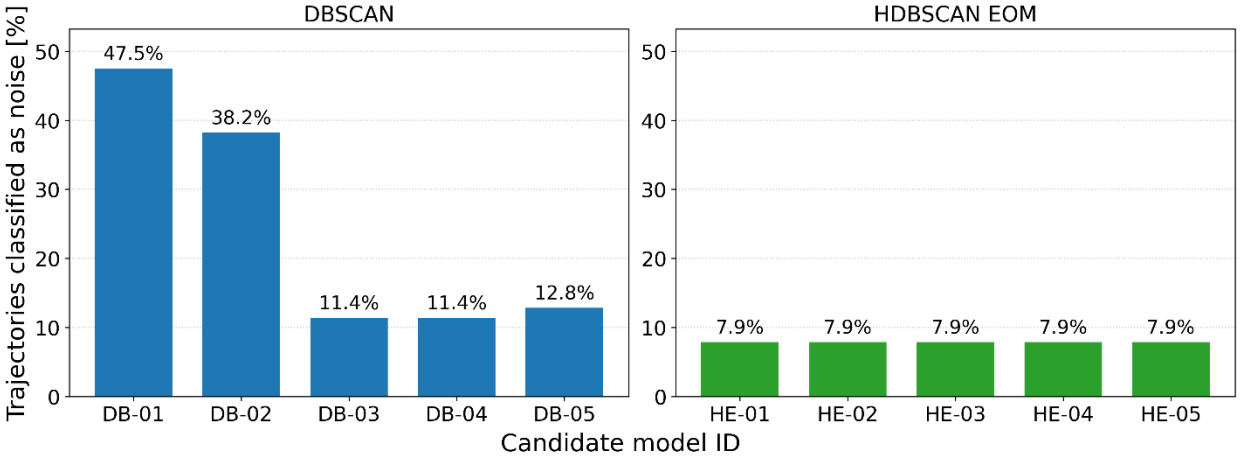


Figure 112: Noise fractions for the shortlisted DBSCAN and HDBSCAN candidate models

Figure 113 presents the corresponding Calinski–Harabasz results. DBSCAN produces the highest values (1,566.23–2,066.75), followed by HDBSCAN, for which all five shortlisted configurations yield a value of 1,241.18. KMeans (321.23–440.59) and full-covariance GMM (319.63–435.07) achieve substantially lower scores. These results indicate that the density-based configurations exhibit stronger within-cluster compactness relative to between-cluster separation according to this metric. However, their higher scores must be considered alongside the smaller number of identified clusters and the exclusion of some trajectories as noise. Consequently, the Calinski–Harabasz results favor DBSCAN and HDBSCAN in terms of internal cluster validity but do not, by themselves, establish their suitability for the intended behaviour-based aircraft categorization application.

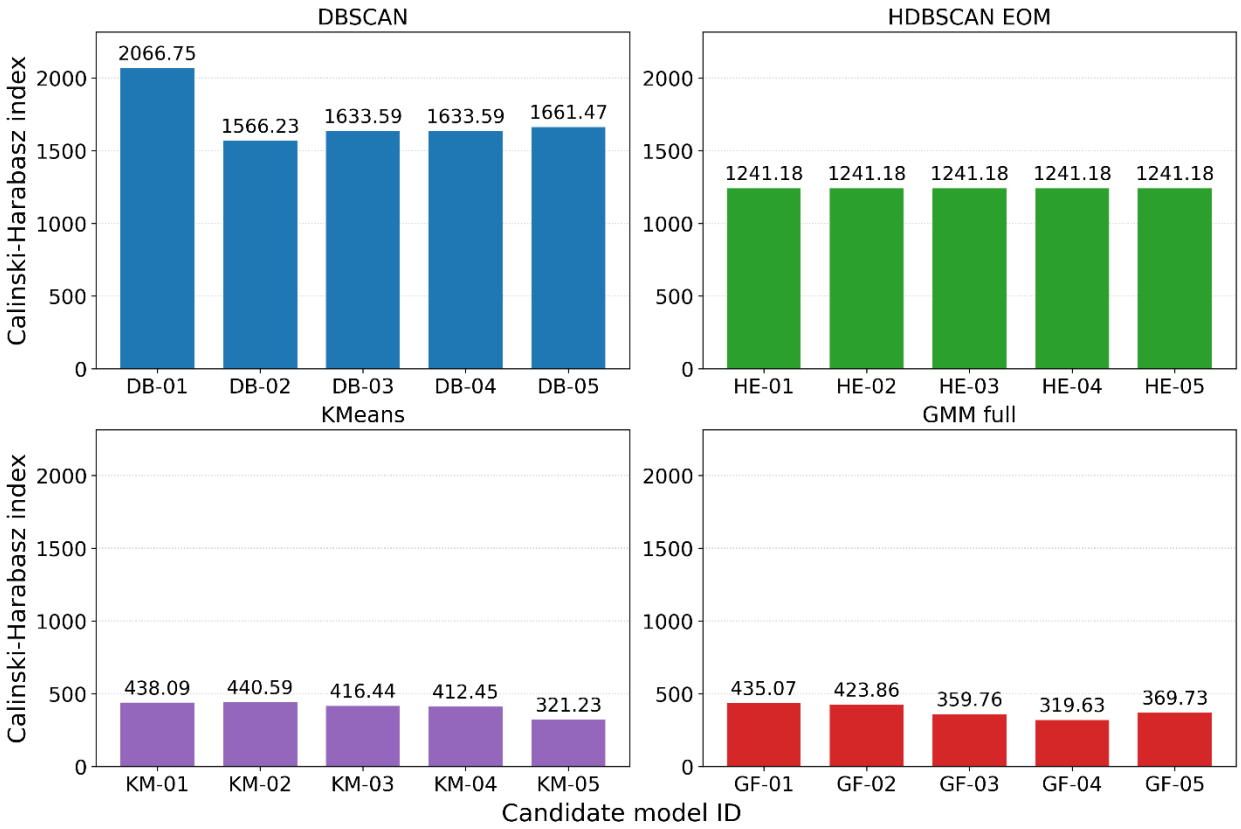


Figure 113: Comparison of Calinski–Harabasz values for the 5 shortlisted candidates from each clustering algorithm

Figure 114 presents the Davies–Bouldin results, for which lower values indicate better within-cluster compactness relative to between-cluster separation. DBSCAN achieves the lowest values (0.20–0.36), followed by HDBSCAN, for which all five shortlisted configurations yield a value of 0.36. KMeans (0.57–0.77) and full-covariance GMM (0.51–0.91) generally produce higher values. These results reinforce the findings from the Silhouette and Calinski–Harabasz metrics, indicating that the density-based configurations exhibit stronger internal cluster validity. However, their favorable scores must be considered alongside their smaller effective cluster counts and the exclusion of some trajectories as noise. Consequently, the Davies–Bouldin results favor DBSCAN and HDBSCAN in terms of cluster compactness and separation but do not, by themselves, establish their suitability for the intended behaviour-based aircraft categorization application.

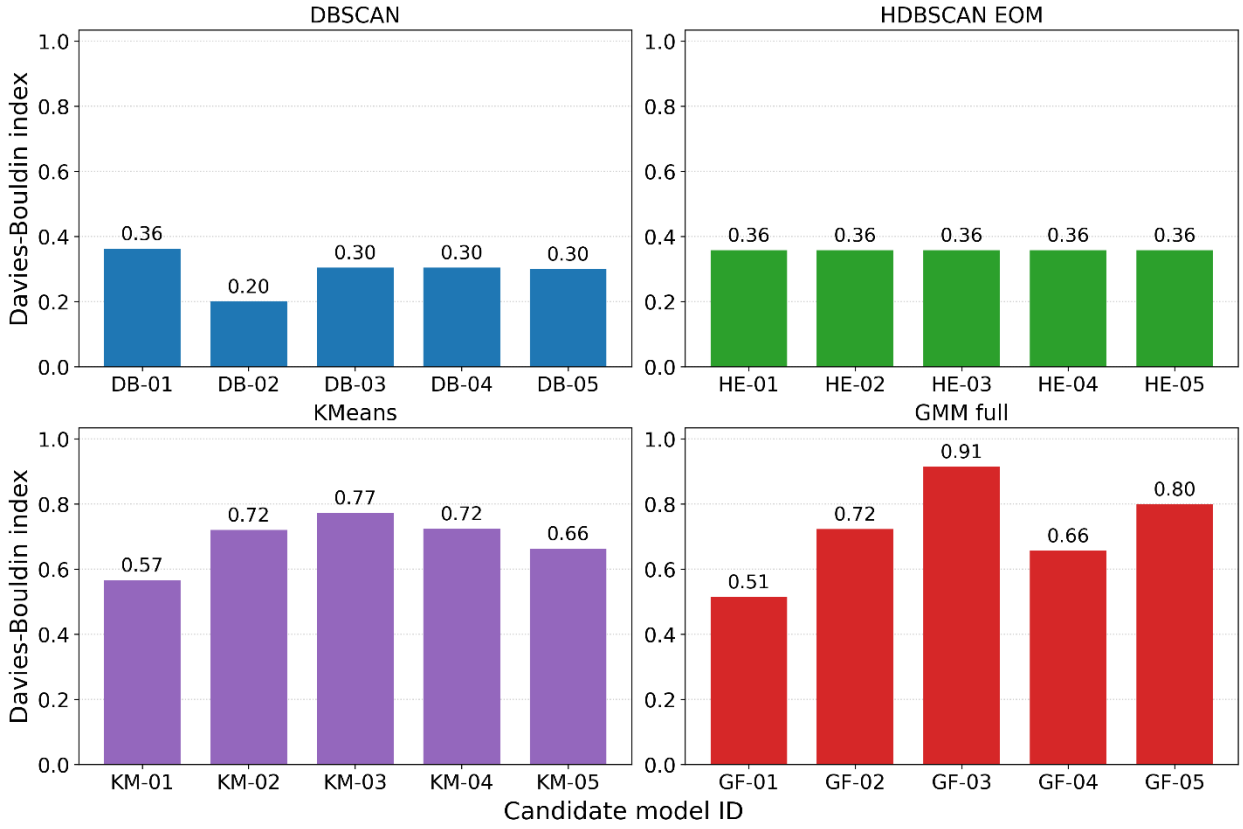


Figure 114: Comparison of Davies–Bouldin values for the 5 shortlisted candidates from each clustering algorithm

The three internal cluster-validity metrics (Silhouette coefficient, Davies–Bouldin index, Calinski–Harabasz index) therefore convey a consistent message: the density-based methods identify particularly compact and well-separated portions of the trajectory dataset. However, they do not establish that those models are the most suitable for aircraft categorization. For this application, the internal metrics must be considered alongside the proportion of trajectories retained, the number of effective clusters, and whether the resulting groups provide a useful representation of both dominant and lower-density aircraft behaviours.

Figure 115 compares the number of effective clusters produced by the shortlisted candidates with the five broad behaviour families identified in the exploratory analysis in Section 7.2. The shortlisted DBSCAN candidates produce three to four effective clusters, while all five HDBSCAN EOM candidates produce three. KMeans spans three to seven clusters, and GMM spans three to eight clusters; both full-assignment methods include solutions that produce four, five, and six clusters. The density-based shortlists therefore tend to under-segment the broad five-family exploratory structure, whereas KMeans and GMM span both sides of the five-family reference and provide a wider range of category granularity. The five-cluster reference is used only as an interpretive benchmark and is not treated as ground truth.

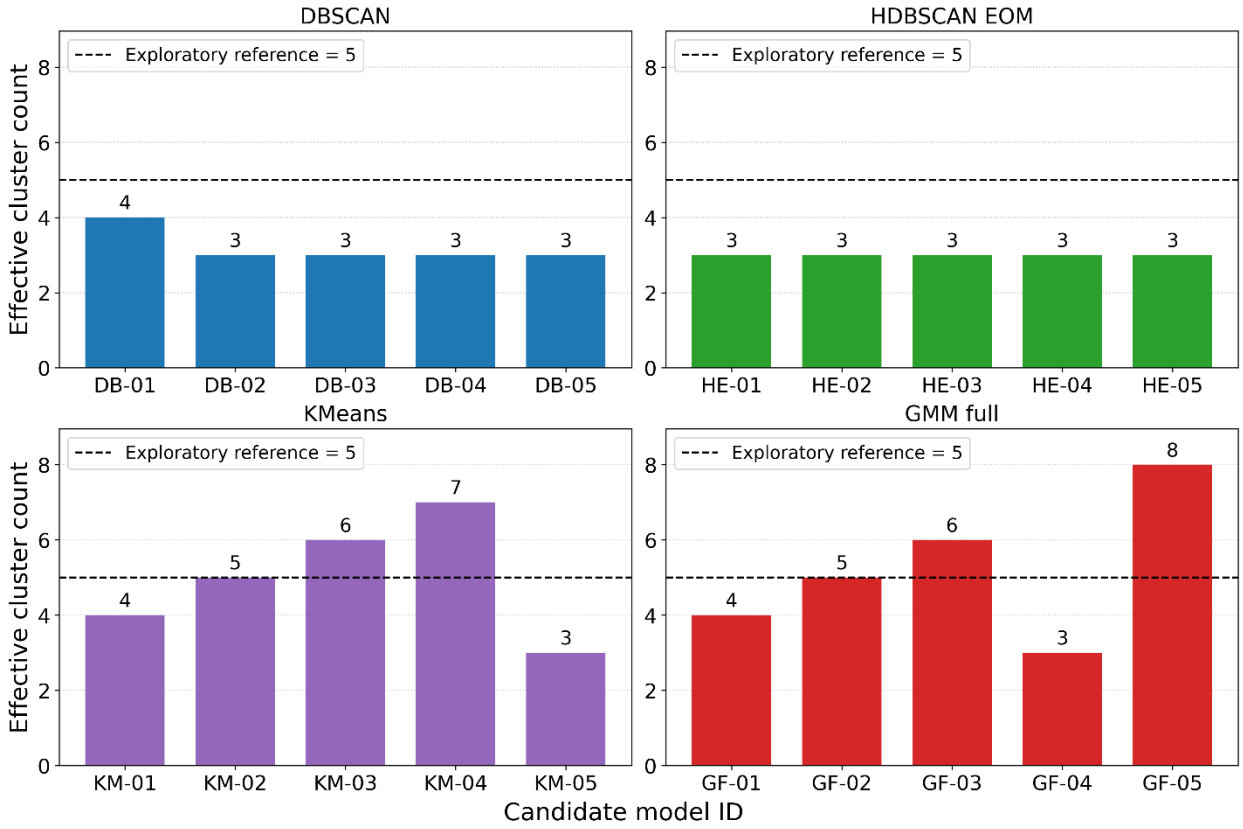


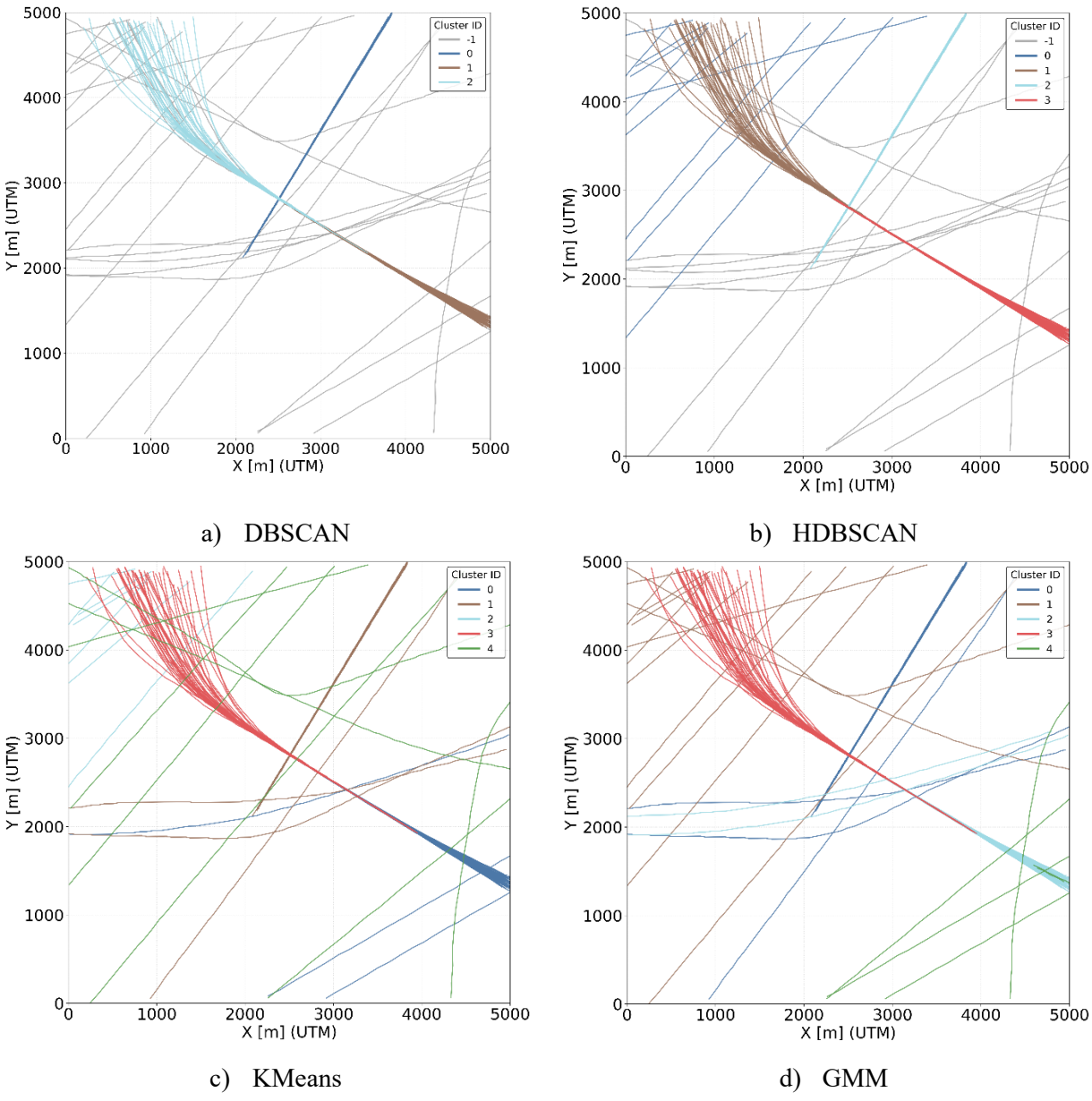
Figure 115: Effective cluster count relative to the five exploratory behaviour families identified in the input dataset

The quantitative comparison identifies a trade-off between internal cluster separation, trajectory coverage, and category granularity. DBSCAN and HDBSCAN achieve stronger internal validity scores for the dense portions of the dataset but leave some trajectories unassigned and generally produce fewer effective clusters. KMeans and GMM provide complete trajectory assignment and allow a broader range of cluster counts, although their internal validity scores are lower. These findings indicate that algorithm selection for the intended aircraft categorization application must consider both the statistical properties of the resulting groups and their ability to represent the operational diversity of the full trajectory dataset.

C.2.3 Representative configuration interpretation

The qualitative comparison examines one representative configuration from each of the four clustering algorithms. These configurations were selected from the broader candidate pool using the quantitative evaluation results, trajectory coverage, effective cluster count, and suitability for operational interpretation. Figures 116–118 compare their resulting cluster assignments through complementary two-dimensional trajectory maps, three-dimensional representations, and aligned altitude profiles. This analysis provides additional evidence of how each algorithm organizes the spatial and temporal behaviour of the aircraft trajectories.

Figure 116 shows substantial differences in the 2D-mapped clusters produced by the four algorithms. DBSCAN isolates three dense terminal-flow clusters and assigns the remaining lower-density trajectories to the grey noise class. HDBSCAN retains more of these trajectories and produces four non-noise clusters. KMeans assigns every valid clustering sample to a cluster and separates several visually distinct trajectory families: Cluster 0 follows the dominant runway 13 departure flow, Cluster 1 follows the runway 22 arrival flow, and Cluster 3 follows the smaller runway 31 departure flow. The remaining KMeans clusters capture lower-density trajectories that cross the LGA-centred region rather than following the dominant runway-aligned paths. GMM also provides complete assignment, but the separation of several lower-density paths is less visually distinct than in the KMeans result.



a) DBSCAN

b) HDBSCAN

c) KMeans

d) GMM

Figure 116: 2D trajectory comparison for the representative DBSCAN, HDBSCAN, KMeans, and GMM configurations

Figure 117 compares the same representative candidates in three dimensions and shows that the differences observed in the 2D trajectory maps also extend to the dataset's altitude structure. DBSCAN isolates three compact groups concentrated primarily within the dense low-altitude terminal traffic and leaves the lower-density trajectories outside these groups as noise. HDBSCAN retains more of the dataset and identifies an additional group containing trajectories that are vertically separated from the principal terminal flows. KMeans assigns every trajectory and separates both the dominant low-altitude groups and the less frequent higher-altitude trajectories into distinct clusters. GMM also provides complete assignment, but the separation between some of the lower-density trajectory families remains less visually distinct. The 3D comparison therefore shows that the candidate algorithms differ not only in horizontal partitioning but also in how they retain and separate the altitude structure of the trajectory dataset.

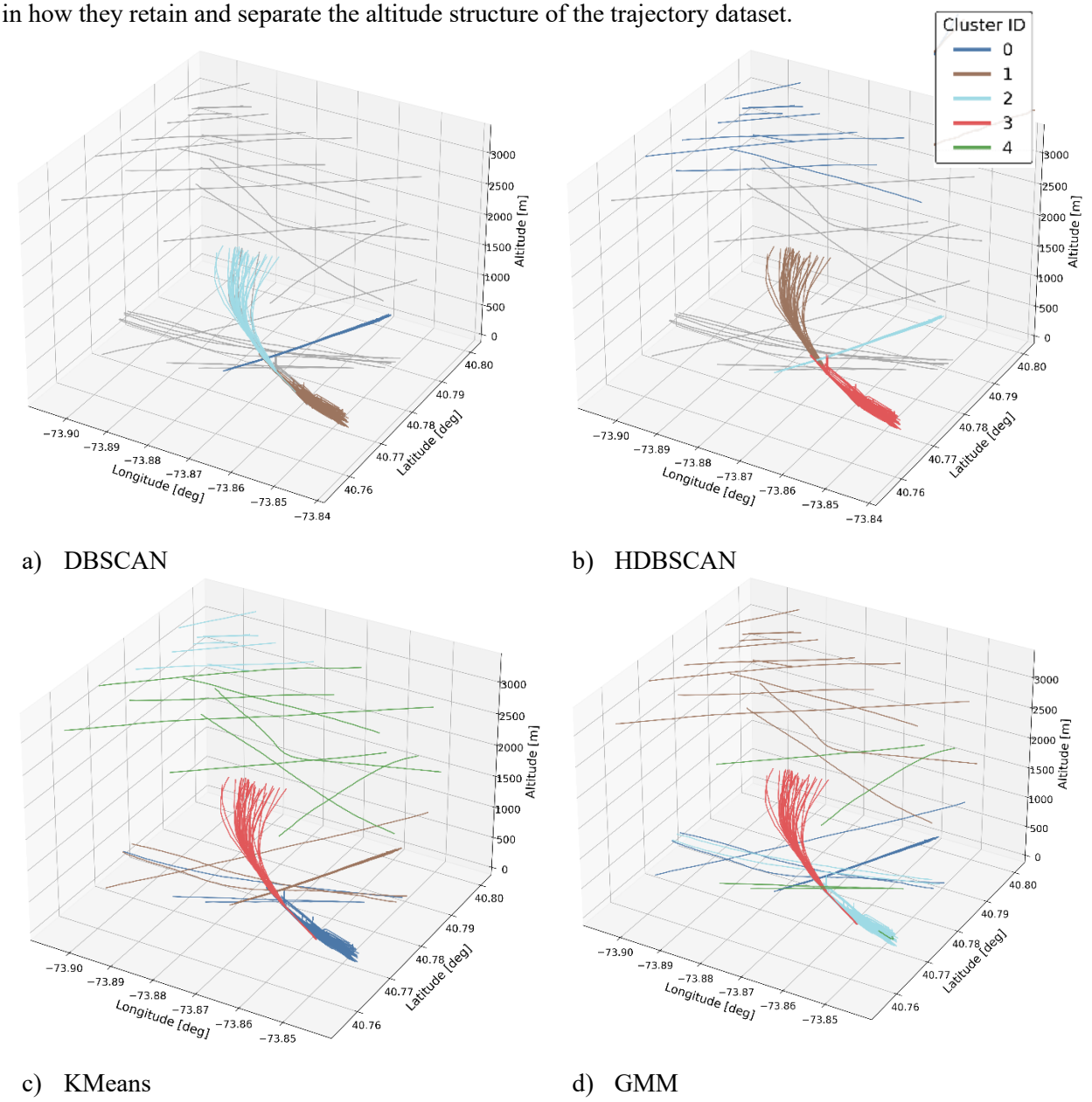
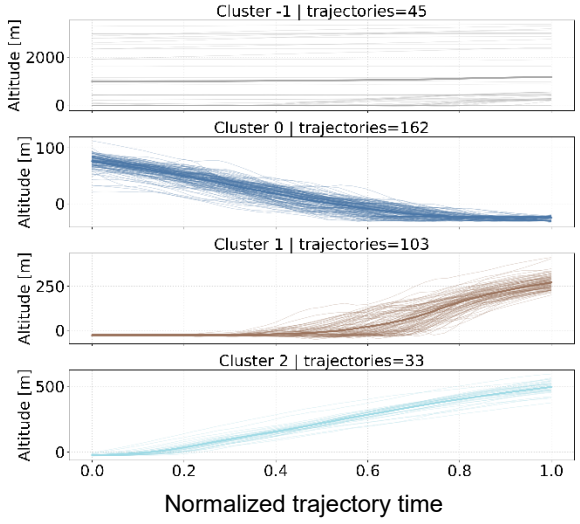
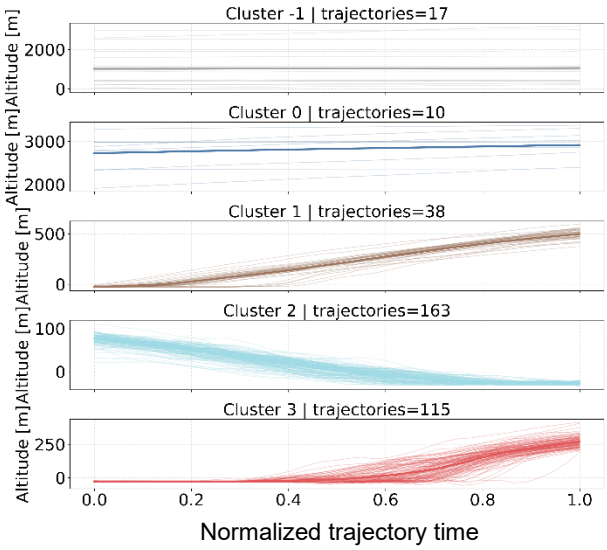


Figure 117: 3D trajectory comparison for the representative DBSCAN, HDBSCAN, KMeans, and GMM configurations

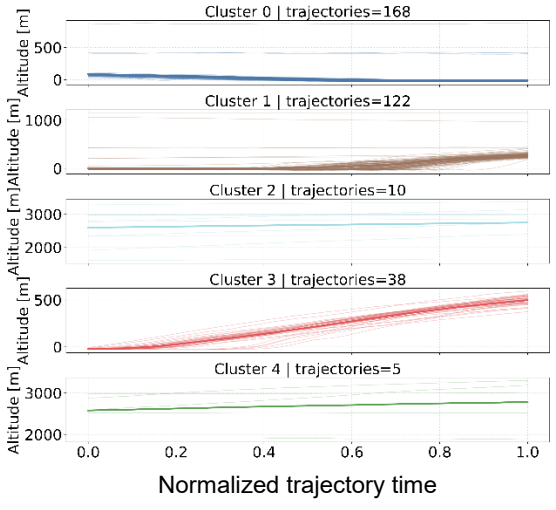
Figure 118 compares the aligned altitude profiles of the same representative configurations, providing a clearer view of how each algorithm separates vertical behaviour over normalized trajectory time. DBSCAN leaves 45 of the 343 trajectories unassigned (13.1%) and primarily retains three compact, low-altitude terminal-aircraft groups. HDBSCAN reduces the unassigned population to 17 trajectories (5.0%) and identifies an additional group of higher-altitude trajectories. KMeans assigns all 343 trajectories to five groups, including a dominant low-altitude group (168 trajectories), two groups exhibiting different climbing profiles (122 and 38 trajectories), and two smaller higher-altitude groups (10 and 5 trajectories). GMM also assigns all trajectories to five groups, identifying a comparable dominant low-altitude group (168 trajectories), two climbing groups (118 and 38 trajectories), and two smaller groups displaying relatively stable flight at different altitudes (13 and 6 trajectories). Although KMeans and GMM both distinguish the main vertical behaviour patterns, they differ in their allocation of the less frequent higher-altitude trajectories. The altitude comparison therefore illustrates the central trade-off observed in the candidate evaluation: strong density-based separation does not necessarily provide the most complete representation of the operational behaviours present in the dataset.



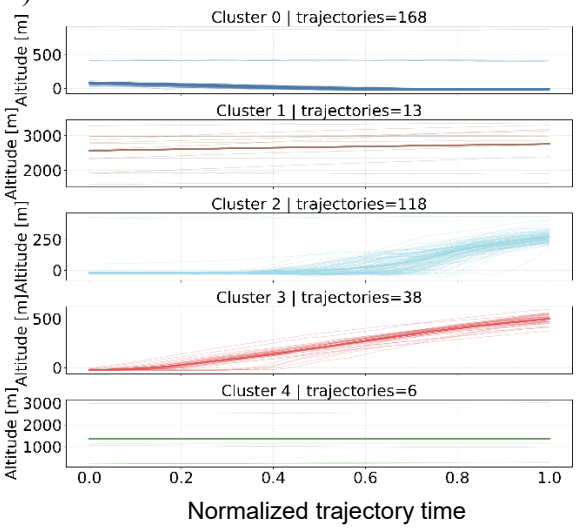
a) DBSCAN



b) HDBSCAN



c) KMeans



d) GMM

Figure 118: Aligned altitude over time comparison for the representative DBSCAN, HDBSCAN, KMeans, and GMM configurations

Together, Figures 116–118 demonstrate that the four clustering algorithms emphasize different properties of the same trajectory dataset. DBSCAN provides strong separation of compact, high-density trajectory groups but leaves a portion of the lower-density trajectories unassigned. HDBSCAN improves trajectory coverage while retaining a density-based representation of the dataset. KMeans provides complete assignment and separates both the dominant terminal-area trajectory groups and the less frequent higher-altitude patterns into distinguishable groups. GMM also provides complete assignment but produces a less directly interpretable decomposition of several lower-density trajectory patterns for the present application. These differences demonstrate why the algorithm-selection decision requires consideration of trajectory coverage and operational interpretability in addition to internal cluster-validity metrics.

Table 32 summarizes the main strengths, limitations, and interpretation of each candidate algorithm for Experiment #3.

Table 32: Summary of the candidate algorithm comparison

Algorithm	Main strength	Main limitation	Interpretation for Experiment #3
DBSCAN	Produces compact and well-separated dense trajectory groups; achieves strong Silhouette, Calinski-Harabasz, and Davies-Bouldin results in the best cases.	Labels 13.1% of the trajectories in the representative solution as noise and tends to under-segment the broader behaviour structure.	Useful for dense-flow detection, but less suitable for full behaviour-based aircraft categorization.
HDBSCAN	Provides the strongest density-based alternative; more consistent than DBSCAN across the tested hyperparameter range; reduces the number of unclustered trajectories; adapts better to heterogeneous density.	Still remains governed by local density and tends to recover fewer broad behaviour families than the full-assignment methods.	Strong density-based comparison model, but not the clearest final behaviour-category model.
KMeans	Assigns every trajectory to a cluster; produces simple, compact, and interpretable categories; performs well near the expected behaviour-family range.	Requires a predefined number of clusters and favours compact partitioning even when the data contain irregular density structure.	Best balance of quantitative performance, trajectory coverage, computational practicality, and operational interpretability for this application.
GMM	Supports probabilistic assignment and more flexible cluster shapes than KMeans; produces plausible results in the four-to-six cluster range.	More sensitive to model specification; more sensitive to the number of components and covariance specification; and less directly interpretable than KMeans for the selected dataset.	Useful comparison model, but does not provide a clearer result than KMeans for the aircraft categorization objective.

C.3 Selection of KMeans for the aircraft categorization application

The results presented in Section D.2 support the selection of KMeans for the behaviour-based aircraft categorization application. Among the four evaluated algorithms, KMeans provides a suitable balance between trajectory coverage, category granularity, and operational interpretability. Its complete trajectory assignment allows both dominant terminal-area flows and less frequent higher-altitude behaviours to be represented, while the resulting clusters remain distinguishable through their spatial and kinematic characteristics.

These properties are particularly relevant to the objective of Experiment #3, which requires the identification and subsequent interpretation of recurring operational patterns without predefined behavioural labels. Although the density-based algorithms achieve stronger internal cluster-validity scores for compact trajectory groups, the KMeans implementation offers a more suitable representation of the broader trajectory dataset for the selected application.

The selected KMeans implementation is consequently used in Chapter 7 to identify and interpret behaviour-based aircraft categories. Its selection reflects the combined quantitative and qualitative evidence presented in this appendix rather than optimization of any single clustering metric. The selection is based on the combined evidence from internal validity metrics, trajectory coverage, behaviour across the tested parameter ranges, computational simplicity, and qualitative trajectory interpretation rather than on the optimum value of any single clustering metric.

The comparison also demonstrates that alternative algorithms remain appropriate for different analytical objectives. Density-based methods, for example, provide advantages when the objective is to isolate the strongest recurrent flows or explicitly identify lower-density trajectories as noise. Appendix D therefore provides the supporting evidence for the KMeans implementation used in Chapter 7 while preserving the broader conclusion of [3] that clustering-algorithm selection should be guided by the interaction between dataset structure and the intended analytical objective.