ABDULRAHMAN ABDULLAH ALHARBI

DEVELOPMENT AND EVALUATION OF DATA ANALYTICS AND MACHINE LEARNING APPROACHES FOR ENHANCED URBAN AIR MOBILITY OPERATIONS

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING

Doctor of Philosophy (PhD)
Academic Year: 2019 - 2023

Supervisor: Ivan Petrunin
Associate Supervisor: Dimitrios Panagiotakopoulos
December 2023

SCHOOL OF AEROSPACE, TRANSPORT AND MANUFACTURING

Doctor of Philosophy (PhD)
Academic Year 2019 - 2023

ABDULRAHMAN ABDULLAH ALHARBI

DEVELOPMENT AND EVALUATION OF DATA ANALYTICS AND MACHINE LEARNING APPROACHES FOR ENHANCED URBAN AIR MOBILITY OPERATIONS

Supervisor: Ivan Petrunin
Associate Supervisor: Dimitrios Panagiotakopoulos
December 2023

This thesis is submitted in partial fulfilment of the requirements for the degree of Doctor of Philosophy

ABSTRACT

Due to their flexibility and general robustness, unmanned aerial vehicles (UAVs), have increasingly been deployed for diverse applications. These include aerial mapping, surveillance, package delivery, and even agriculture. Increased employment, however, has also entailed new demands for smart, nimble and effective UAV traffic-management systems, particularly in urban areas. If numerous, fully automized UAVs are to be flown frequently, and beyond the visual line of sight (BVLoS), then efficient unmanned traffic management (UTM) is essential, not least as UAV traffic will inevitably become denser. In future, indeed, air-traffic management will also be more complex, and airspace more crowded, as the sheer volume of UAVs continues to rise. Consequently, UTM will require swift, efficient decision-making mechanisms.

Important challenges also remain in terms of machine-learning algorithm verification, these stemming primarily from a lack of explicability and transparency. Given that traditional safety mechanisms are unequal to the tasks involved, this has been an inhibiting factor in the integration of UAVs into very low-level (VLL) airspace.

This thesis develops a data-analytics framework to analyze simulated historical data and characterize traffic-flow patterns in UTM airspace. The framework enhances risk analysis and improves trajectory planning across various airspace regions. It considers all dynamic parameters, such as extreme weather, emergency services, and dynamic airspace structures. Furthermore, and to meet the critical need for accurate congestion prediction in UAS traffic-flow management (UTFM), this study uses state-of-the-art machine learning techniques to integrate air traffic-flow prediction with the intrinsic complexity metric. In this study, air-traffic congestion analysis and prediction will be addressed via a deep-learning methodology, within a UTM context, across a timeframe of three minutes. The proposed model is distinct from approaches that would focus on the more conventional issues of conflict detection, conflict resolution and trajectory prediction.

In addition, this thesis proposes a tailored solution to the needs of demand-and-capacity-management (DCM) services. This solution deploys a transparency-based methodology, with a fusion of both black-box and explainable, white-box models. It generates, therefore, an intelligent system that can be both explicable and reasonably comprehensible. The results show that the advisory system will be able to indicate the most appropriate regions for UAV operations, while increasing UTM airspace availability by more than 23%.

Keywords:

Deep Learning, Explainable Artificial Intelligence, Long short-term memory (LSTM) networks, Low-Altitude Airspace Operations, Machine learning, Traffic Flow Management, Trajectory data analytics, Unmanned aerial vehicle (UAVs), Unmanned aircraft traffic management (UTM).

ACKNOWLEDGEMENTS

This work is dedicated to my family, who offered invaluable support throughout this research.

Many individuals have provided important assistance during this PhD project, the completion of which has been “life changing” in the most positive way. First, I would like to extend my sincere thanks to Dr Ivan Petrunin, my supervisor, for his unstinting encouragement during my time at Cranfield University. It is a huge advantage to have a supervisor who not only responds quickly and thoughtfully to one’s questions, but who takes a deep and manifest interest in the project itself. From first to last, Dr Ivan distinguished himself by his empathy, wisdom and practical support. This thesis would have been impossible without his unfailing encouragement and invaluable feedback. I would also like to thank my associate supervisor, Dr Dimitrios Panagiotakopoulos. During my time as his student, Dimitrios provided numerous insights and highly valuable advice.

Above all, however, my gratitude applies to Waad, my wife, for her continued and patient support. In my absence, she was obliged to maintain a home and family almost single-handed, and she rose to this task with typical spirit and generosity. Further thanks must be given to my mother and brothers, who encouraged me to pursue my dreams, and who invariably believed in my capacity to realize them. Friends and colleagues at Cranfield, moreover, provided both practical and emotional support during my academic journey.

Finally, I am pleased to acknowledge the funding of the Royal Saudi Air Force, which was an essential component in the completion of this PhD project.

TABLE OF CONTENTS

ABSTRACT i
ACKNOWLEDGEMENTS iii
TABLE OF CONTENTS v
LIST OF FIGURES viii
LIST OF TABLES xii
LIST OF EQUATIONS xiii
LIST OF ABBREVIATIONS xiv
1 GENERAL INTRODUCTION 17
1.1 Background Context 17
1.2 Motivation 20
1.3 Aim and Objectives 24
1.4 Research Design 25
1.5 Thesis Structure 28
1.6 Contribution to Knowledge 30
2 UTM SIMULATION OVERVIEW AND IDEALISED TRANSPORT MODEL REQUIREMENTS 33
2.1 Introduction 33
2.2 Idealized Transport Model Requirements 33
2.2.1 Idealized Transport Model Use Case Diagram 40
2.2.2 Assumptions and Considerations for Case Study Implementation 42
2.3 Overview of Simulation-based Tools for The UTM System 47
2.4 Path Planning Optimization 52
2.4.1 Path Planning Techniques 53
2.5 Rule-Based Conflict Management for UTM Scenarios 58
2.5.1 Review of Conflict Detection and Resolution Methods in UTM 58
2.5.2 Deconfliction Methodology 63
2.5.3 Simulation Scenarios 65
2.5.4 Deconfliction Methodology 70
2.5.5 Application of Deconfliction 73
2.5.6 Deconfliction Methodology Results Discussion 77
2.5.7 Conclusion and Future Recommendations 79
3 MODELING AND CHARACTERIZATION OF TRAFFIC FLOW PATTERNS AND IDENTIFICATION OF AIRSPACE DENSITY FOR UTM APPLICATION 80
3.1 Introduction 80
3.2 Related Work 83
3.2.1 Overview of Trajectory Data Mining Techniques 83
3.2.2 Flight Data Analysis and Traffic pattern Identification 86
3.3 Methodology 90
3.3.1 Simulation Setup 90

3.3.2 Weather Effects' Implementation 93
3.3.3 Framework For Characterization of Air Traffic Pattern 94
3.4 Simulation Scenario and Results 99
3.4.1 Scenario 1 – NFZ With no Weather Constraint and Dynamic Obstacles 101
3.4.2 Scenario 2 – Static Airfield and Dynamic Recreational Area with Extreme Weather Conditions 101
3.4.3 Scenario 3 – Dynamic Airfield and Static Recreational Area with Different Weather Condition (Rain and Wind) 102
3.5 Airspace Traffic Flow Patterns, Characterization and Density discussion 104
3.5.1 Identification of UTM Air Traffic Patterns 104
3.5.2 Statistical Analysis of Clusters and Valid Trajectories 115
3.5.3 UTM Air Traffic Flow Characterization 116
3.5.4 UTM Airspace Congestion Analysis 130
3.6 Conclusion and Future Work 135
4 DEEP-LEARNING ARCHITECTURE FOR UAV TRAFFIC DENSITY PREDICTION 138
4.1 Introduction 138
4.2 Related Work 142
4.3 Methodology 147
4.3.1 Data Generation and Simulation Scenarios 147
4.3.2 UAVs States Formulation 151
4.3.3 Computation of Complexity Metric: Spatio-Temporal Correlation 152
4.3.4 Implementation of Encoder-Decoder LSTM Model 154
4.3.5 Model Training 157
4.4 Results and Discussion 160
4.4.1 Prediction Result 160
4.4.2 Justification of the Selection of the Look-Ahead for Prediction 166
4.4.3 Discussion on the Generalization of the Proposed Model 170
4.4.4 Comparison Between Existing Approaches and The Proposed Model 172
4.4.5 Comparison Between Existing Approaches and The Proposed Model: Performance vs. Time 175
4.4.6 Comparison Between Existing Approaches and The Proposed Model: Computation Speed 176
4.5 Conclusions and Future Work 178
5 ASSURING SAFE AND EFFICIENT OPERATION OF UAV USING EXPLAIMABLE MACHINE LEARNING 180
5.1 Introduction 180
5.2 Literature Review and Background 184
5.2.1 The Challenges of Certifying AI 185

5.2.2 State-of-the-Art ML: Methods of Interpretability 188
5.3 Proposed Advisory System Framework 194
5.3.1 Overall Framework 194
5.4 Results and Discussion 197
5.4.1 UTM Congested Subzones: Identification and Area Distribution 197
5.4.2 Analysis and Design of Explainability for The DCM Advisory System 208
5.4.3 Advisory-System Efficiencies for Capacity and Safety 218
5.5 Conclusion and Future Work 220
6 GENERAL DISCUSSION 223
6.1 Introduction 223
6.2 Simulation Development 223
6.3 UTM Airspace Traffic Pattern Characterization 227
6.4 Traffic Flow Prediction for UTM Application 228
6.5 Explainable AI for DCM Services 229
7 CONCLUSIONS AND FURTHER RESEARCH 233
7.1 Addressing the Research Aim and Objectives 233
7.2 Future Work 237
References 240

LIST OF FIGURES

Figure 1-1 Thesis workflow diagram 26
Figure 2-1 Use Case Model Design Semantics [73] 41
Figure 2-2 Idealized Transport Model Use Case 41
Figure 2-3 The requirements of the UTM Simulator 52
Figure 2-4 Flowchart of a PSO algorithm [128] 57
Figure 2-5 Conflict management layers in ATM and UTM [130] 59
Figure 2-6 Rule-based conflict management for UTM scenarios 64
Figure 2-7 First hour simulation - conflict heat map 66
Figure 2-8 Second hour simulation - conflict heat map 67
Figure 2-9 Third hour simulation - conflict heat map 68
Figure 2-10 Number of conflicts and agents in the sky from 8AM to 11AM 68
Figure 2-11 Conflicts per number of flights 69
Figure 2-12 First stage of strategic deconfliction 71
Figure 2-13 Second stage of strategic deconfliction 72
Figure 2-14 Third stage of strategic deconfliction 73
Figure 2-15 Rerouting Example: Confliction case 74
Figure 2-16 Rerouting Example: Deconfliction case 75
Figure 2-17 Conflict resolution by process 76
Figure 3-1 Simulation environment created by Air Map 91
Figure 3-2 Overview of the UTM airspace characterization approach 96
Figure 3-3 DBSCAN clustering [155] 98
Figure 3-4 Simulation scenario 9:00 am to 10:00 am 101
Figure 3-5 Simulation scenario 10:00 am to 11:00 am 102
Figure 3-6 Simulation scenario 11:00 am to 12:00 pm 103
Figure 3-7 Simulation-scenario environment for the ideal case scenario 106
Figure 3-8 Scenario-0 DBSCAN results 107
Figure 3-9 Scenario-0 DBSCAN detected mission cluster plots 108
Figure 3-10 Scenario-0 Detected combined mission patterns 108

Figure 3-11 Scenario-0 Trajectory count per cluster (%) 109
Figure 3-12 Scenario-1 DBSCAN results 109
Figure 3-13 Scenario-1 DBSCAN detected mission cluster plots 110
Figure 3-14 Scenario-1 Detected combined mission patterns 110
Figure 3-15 Scenario-1 Trajectory count per cluster (%) 111
Figure 3-16 Scenario-2 DBSCAN results 111
Figure 3-17 Scenario-2 DBSCAN detected mission cluster plots 112
Figure 3-18 Scenario-2-Detected combined mission patterns 112
Figure 3-19 Scenario-2 trajectory count per cluster (%) 113
Figure 3-20 Scenario-3 DBSCAN results 113
Figure 3-21 Scenario-3 DBSCAN detected mission cluster plots 114
Figure 3-22 Scenario-3- Detected combined mission patterns 114
Figure 3-23 Scenario-3 Trajectory count per cluster (%) 115
Figure 3-24 The concept of DFC measurement 117
Figure 3-25 Scenario-0: Covid-19 sample delivery (left) and delivery services (right) mission polygons with Centroids (*). 121
Figure 3-26 Scenario-0: NHS and delivery services (DS) missions DFC Statistical Distribution 122
Figure 3-27 DTCM & TTCM distribution for Scenario-0 122
Figure 3-28 Scenario-1: Covid-19 sample delivery (left) and delivery services (right) mission polygons with Centroids (*). 123
Figure 3-29 Scenario-1: NHS and DS missions DFC Statistical Distribution .. 124
Figure 3-30 DTCM & TTCM distribution for Scenario-1 125
Figure 3-31 Scenario-2: Covid-19 sample delivery (left) and delivery services (right) mission polygons with Centroids (*) 126
Figure 3-32 Scenario-2: NHS and DS missions DFC Statistical Distribution .. 127
Figure 3-33 DTCM & TTCM distribution for Scenario-2 128
Figure 3-34 Scenario-3: NHS (left) and delivery services (right) mission polygons with Centroids (*) 128
Figure 3-35 Scenario-3: NHS and DS missions DFC Statistical Distribution .. 129
Figure 3-36 DTCM & TTCM distribution for Scenario-3 130

Figure 3-37 Airspace Congestion identification for scenario 1..... 132

Figure 3-38 Airspace Congestion identification for scenario 2..... 133

Figure 3-39 Airspace Congestion identification for scenario 3..... 134

Figure 4-1 Simulation-scenario environment, with UAV missions map 149

Figure 4-2 Scenario 3: 200 UAVs with extreme weather effects 151

Figure 4-3 The dynamic system and its eigenvalues: for dynamic-system matrix AU; the eigenvalue loci for two typical situations, namely, convergent (a) and divergent (b). One should note that the real component of the eigenvalues is negative only in the case of convergent trajectories. Meanwhile, the 'neighborhood' of a reference aircraft (for the computation of the complexity metric) is indicated by (c). (Adapted from the work of García et al. [261]) 154

Figure 4-4 LSTM for time series prediction—general architecture [263] 155

Figure 4-5 Overview of the proposition for the UTM traffic flow prediction model 156

Figure 4-6 The process of data sliding window for creating a time series data 159

Figure 4-7 First scenario: Actual and predicted complexity in the airspace 160

Figure 4-8 The difference between predicted and true congestion—Scenario 1 161

Figure 4-9 Percentage residual complexity—Scenario 1..... 162

Figure 4-10 Scenario 2: Actual and predicted complexity in the airspace 163

Figure 4-11 The difference between predicted and true congestion—Scenario 2 164

Figure 4-12 Percentage residual complexity—Scenario 2..... 164

Figure 4-13 Scenario 3: Actual and predicted complexity in the airspace 165

Figure 4-14 The difference between predicted and true congestion—Scenario 3 166

Figure 4-15 Percentage residual complexity—Scenario 3..... 166

Figure 4-16 Comparison of Input Sample Windows and Look-Ahead Times for Predicting Traffic Congestion in UTM 168

Figure 4-17 Normalized RMSE on training data (Scenario 1) and validation data (Scenarios 2 and 3) 171

Figure 4-18 Comparison of different prediction approaches..... 174

Figure 4-19 Prediction vs. time for (a) shallow NN, (b) regression, and (c) proposed model 175

Figure 4-20 Computation timing for shallow NN, regression, and proposed model 177

Figure 4-21 Prediction timings percent increase 177

Figure 5-1 Classification of explainable AI methods [279] 190

Figure 5-2 The explainable machine-learning model for the safe and efficient operation of UAV 196

Figure 5-3 Congestion-cluster levels for Scenario 3: (a) lowest level, (b) lower level, (c) medium level, (d) higher level, and (e) highest level 200

Figure 5-4 Area distribution per level, per cluster: Scenario 3 201

Figure 5-5 Cumulative area distribution per level: Scenario 3 201

Figure 5-6 Percentage congestion distribution for all three scenarios: a) first, (b) second, and (c) third scenario 204

Figure 5-7 Traffic flow distribution and airspace capacity: Scenario 1 205

Figure 5-8 Traffic-flow distribution and airspace capacity: Scenario 2 206

Figure 5-9 Traffic-flow distribution and airspace capacity: Scenario 3 206

Figure 5-10 Rule-based decision tree for DCM services 212

Figure 5-11 Decision-support results: Scenario 1 214

Figure 5-12 Decision-support results: Scenario 2 214

Figure 5-13 Decision-support results: Scenario 3 215

Figure 5-14 Explanation by example for Scenario 1, lowest congestion level 217

Figure 5-15 Explanation by example for Scenario 1, highest congestion level 218

Figure 5-16 Capacity gain (CG) and safety gain (SG) for all three scenarios. 220

LIST OF TABLES

Table 2-1 Qualitative comparison of path planning algorithms 56

Table 3-1 The scale of weather classification used in this study 94

Table 3-2 The description and schedule for UAVs missions 100

Table 3-3 Technical parameters of UAVs 100

Table 3-4 DBSCAN cluster's parameters for simulation scenarios 104

Table 3-5 Valid mission & noisy clusters' percentages 116

Table 3-6 Number & type of missions detected with DBSCAN 116

Table 4-1 The description and schedule for UAVs missions 149

Table 4-2 Technical parameters of UAVs used during the simulation scenarios 150

Table 4-3 List of the proposed model's parameters 157

Table 4-4 List of the proposed model's hyperparameters values 158

Table 5-1 Congestion levels 195

Table 5-2 DBSCAN parameters per congestion level 198

Table 5-3 Statistics for area utilization ($km^{2)}$ per congestion level, for all scenarios 202

Table 5-4 Lowest- and highest-congestion clusters: cumulative areas (km^2) 203

Table 5-5 Capacity gain: a comparison 207

Table 5-6 Methodology transparency scoring 209

Table 5-7 Advisory-system transparency 210

Table 5-8 Advisory-system efficiency 219

LIST OF EQUATIONS

(3-1)..... 93
(3-2)..... 93
(3-3)..... 94
(3-4)..... 94
(3-5)..... 97
(3-6)..... 97
(3-7)..... 119
(3-8)..... 119
(3-9)..... 119
(3-10)..... 119
(3-11)..... 120
(3-12)..... 120
(4-1)..... 152
(4-2)..... 152
(4-3)..... 152
(4-4)..... 153
(4-5)..... 153
(4-6)..... 153
(4-7)..... 154
(4-8)..... 156
(4-9)..... 159
(4-10)..... 171
(4-11)..... 173
(4-12)..... 173
(4-13)..... 173
(5-1)..... 219
(5-2)..... 219

LIST OF ABBREVIATIONS

UAV	Unmanned Aerial Vehicle
BVLOS	Beyond-Line-of-Sight
UTM	Unmanned Traffic Management
VLL	Very Low-Level
UAS	Unmanned Aircraft Systems
UTFM	UAS Traffic Flow Management
DCM	Demand-and-Capacity-Management
LSTM	Long Short-Term Memory
ATM	Air Traffic Management
NAS	National Airspace System
HTOL, VTOL	Horizontal take-off and Vertical or Hybrid
UAM	Urban Air Mobility
ATC	Air Traffic Control
AI	Artificial Intelligence
UTFP	UTM Traffic Flow Prediction
ATFCM	Air Traffic Flow and Capacity Management
NFZs	No-Fly Zones
PSO	Particle Swarm Optimization
ATFP	Air Traffic Flow Prediction
DBSCAN	Density-Based Spatial Clustering of Applications with Noise
ML	Machine Learning
LDS	Linear Dynamical System
XAI	Explainable Artificial Intelligence
ATFM	Air Traffic Flow Management
CDR	Conflict Detection and Resolution
ICAO	International Civil Aviation Organization
DFC	Distance from Centroid
DTCM	Distance to Complete Mission
TTCM	Time to Complete Mission
KDE	Kernel Density Estimation
CNN	Convolutional Neural Network
IFR	Instrument Flight Rules
VFR	Visual Flight Rules
EASA	European Union Aviation Safety Agency

LIME	Local Interpretable Model-agnostic Techniques
SHAP	Shapley Additive Explanations

1 GENERAL INTRODUCTION

1.1 Background Context

The Unmanned Aerial Vehicle (UAV) presents an immense opportunity for commercial and industrial activities, due to its robustness and flexibility. The global market for UAVs is predicted to expand at a rate of 15.88% by 2026 [1], and it is anticipated that demand for different services, such as delivery, surveillance, farming activities, and aerial photography will also increase [2]–[6]. The proposed new vehicles will be cheaper than traditional helicopters [7], less noise polluting [8], and generally more efficient [9]. These advantages stem from their use of lightweight materials, multiple propellers and motors, and electric engines. Operations will cover both rural and urban areas [10], while smaller industry separation standards will flow from competition (among multiple operators) for the same finite space [11]. Operations will probably implicate electric vehicles, remotely manned or unmanned, with vertical take-off and landing capability.

The degrees of mobility and autonomy that unmanned air traffic, and its related elements, require are, however, not supportable via conventional systems of Air-Traffic Management (ATM) [12]. The integration of urban operations with current National Airspace System (NAS) operations is being prevented by a range of factors. These include: 1) higher operational density, 2) lower operational altitudes, 3) increasing volumes of operation, and 4) the disparate performance characteristics of different vehicles and operators [13]. Absent a suitable, highly automated management system, dense drone operations may imperil the safety, not to mention the privacy, of citizens [14]. Therefore, various platform architectures, practical demonstrators, conceptual frameworks, and methodologies have been developed to address the complex challenge of accommodating unprecedented levels of management for civilian drone traffic. One of these initiatives is a new US system for Unmanned Aircraft Systems Traffic Management (UTM), developed at the Ames Research Center of NASA [15]. The federated services that this system comprises have the capacity (e.g.) to support the secure, and safe, operations of UTMs. In turn, this implicates areas

such as communications, weather, flight-operational planning, and separation. Meanwhile, Single European Sky ATM Research Joint Undertaking (SJU) developed its own model, named U-Space (2017) [16]. The latter, as an enabling framework, can accommodate all forms of mission, such as flight sorties in the vicinity of airports. It also facilitates highly automated, complex drone operations, and it can do so within every type of operational setting. Meanwhile, drone traffic-management frameworks such as the Japanese UTM Consortium (JUTM) and the Chinese Civil UAS Operation Management System (UOMS) are broadly similar to the paradigms of U-Space or UTM [17].

The Uber examination of electric Vertical Take-off and Landing (eVTOL) air-taxis, with particular reference to urban settings, provides a further instance of industry-oriented research. Here, the focus of interest lies primarily with traffic control for point-to-point, fixed-based Vertiports, certification, and eVTOL platform performance [18]. A broader view of drone capabilities, meanwhile, is evinced by the Airbus “Blueprint for the Sky.” This comprises a harmonized, shared, and future-proofing approach to airspace, safety, self-management and self-piloting. Within the eponymous blueprint, types of traffic activity and environment provide the structuring criteria for airspace, examples being free-route, fixed-route and basic-flight structures, as well as corridors [19].

The concepts of Unmanned Traffic Management (UTM) [20] and Urban Air Mobility (UAM) [21] have been proposed, in the studies cited above, in order to facilitate safe and efficient aerial-vehicle operations. It is emphasized in these studies, moreover, that UTM and UAM systems have changed the paradigm of existing air-traffic systems, while posing challenges to aircraft safety and efficiency. These challenges stem from a variety of factors, including divergent vehicle characteristics; increased density of vehicular operations, urban terrain environments, and complex atmospheric conditions in lower airspace. A question thus arises: how, under these challenging conditions, may one handle high-density operations safely and efficiently?

The current UTM system is in its preliminary stages and is unable to handle the anticipated increase in traffic from large UAV volumes and densities [22]. While

expanding conventional air traffic management (ATM) systems for manned aircraft is a potential solution, policymakers recognize that these systems are insufficient to handle the expected volume and diverse nature of UAS traffic [23]. Moreover, traditional ATM systems may not effectively address UAS safety and management concerns, particularly in urban areas where UAS operate in close proximity to people and buildings [24]. Additionally, limited social acceptance of UAS operations in such environments remains a challenge [25]. Although segregated airspace initially addresses UAS safety concerns, it fails to support future integration with manned aviation and high-density UAS operations [26].

Jiang et al. [23] suggest that the UTM system can draw from principles derived from ATM, despite differences in control methodologies, maneuvering techniques, functional elements, operational ranges, and constraints. However, obstacles to wider UAS deployment on a larger scale include the lack of suitable support functions, inadequate operational parameters, and a scarcity of appropriate procedures. Notable differences between UAS and manned aviation include the unknown performance characteristics of UAS as a relatively new technology, the absence of human pilots for vehicle detection and collision avoidance, limitations on carrying heavy or power-intensive components for small, unmanned aircraft systems (sUAS), and variations in separation imperatives and standards compared to traditional aviation. In this context, the primary risks involve property, people on the ground, and manned aerial vehicles in the vicinity [27].

Given the anticipated demands of future transport, it is essential for the UTM system to be capable of enabling optimal Urban Air Mobility, thereby freeing logistics and human traffic from the limitations of congested highways. Achieving integration between manned and unmanned airspace, as well as ensuring effective coordination between the two, necessitates the implementation of seamless beyond-visual-line-of-sight (BVLOS) functionality at both higher and Very Low Level (VLL) altitudes. Consequently, there is a need to design a sustainable UTM systemic paradigm that is interoperable, fully autonomous, and compatible with regulatory frameworks.

In this context, the recent success of data sciences, specifically data-analytic and machine-learning techniques, in the areas of perception, planning, and control [28], render these suitable for application towards the safe and efficient operation of UAVs in the UTM domain. There are, in fact, many applications of data analytics that may bring improvements in different aspects of air-traffic management. These include air-traffic flow extraction [29], aircraft performance-parameter prediction [30], flight-delay prediction [31], taxi-out time prediction [32], and trajectory prediction [33]. The review conducted in Chapters 3, 4, and 5, meanwhile, demonstrates that four main categories may be discerned in terms of AI within ATM. These categories define the global purposes of the application: prediction (e.g., prediction of the traffic), optimization/automation (avoiding conflict, optimizing a trajectory), analysis (e.g., assessing the workload of an Air Traffic Controller, or ATCo), and modelling/ simulation (e.g., simulating the air traffic of a specific region of airspace). The review also demonstrates that data analytics are applied to virtually all the more difficult stages of the ATM domain. Despite this, academic coverage, in the context of UTM, has been meager. This stems both from a lack of research publications, and greater focus being allotted to the implementation and control approaches of UAVs [34]. Moreover, a considerable barrier in data analysis, vis-à-vis UTM, is the absence of a common and shared database to collect real-time UAV flight-operation data. One possible developmental direction is to use simulation data [35].

Vast quantities of data are now captured and stored, following the rapid development of databases, communications technologies, and the Internet of Things (IoT) [36]. Consequently, new-generation air-transport paradigms can be rendered far safer, more predictable, and more efficient by the power latent within data-mining techniques and machine learning.

1.2 Motivation

This study was motivated by the increasing need for efficient, optimized and data-driven machine-learning approaches toward airspace monitoring, airspace design and traffic-flow management in the UTM domain. Moreover, the absence of a pilot on board the UAV introduces serious concerns related to flight safety.

These challenges pertain to the detection and avoidance of dangerous situations, management and control, communication with the air-traffic controller, and the prevention of accidental or unlawful interference [37]. Thus, safety and capacity are two major challenges that the UTM community must address in years to come.

Since most future UAV flying operations will take place in densely and dynamically populated urban areas, there is a strong imperative to study and characterize UTM traffic-flow behaviors covering entire routes. It is also necessary to evaluate UTM capacity estimations along with safety aspects. Moreover, the effects of dynamic factors such as adverse weather, emergency operations, and static and dynamic obstacles need to be studied, to find possible ways to reroute UAV flights through air traffic-flow patterns. The present study will explore opportunities to deploy basic artificial-intelligence (AI) technological applications, to facilitate more efficient and realistic flight planning, and to assist airspace flow management, potentially resulting in higher safety, capacity, and efficiency of airspace.

The principal attraction of a method for detecting anomalies, in the context of air-traffic operations, is the opportunity automatically to identify a manageable volume of significant events, and to do so amid high volumes of trajectories. A safety expert, focusing on these event subsets, will subsequently undertake additional risk assessments. Conversely, hotspots may be generated by deconfliction measures, and by other frequently occurring significant events, within a given region of airspace. In turn, this creates greater complexity and, potentially, greater threats to safety. Thus, the accurate analysis of UTM operational data can play a significant role in improving many vital aspects of UTM, such as safety enhancement, flight efficiency, or delay reduction. From the literature review, it is notable that data analytics are applied in almost all the challenging phases of the ATM domain. Since this topic is quite novel and recent, there is a lack of research publications in the UTM context: most of the existing research has ignored data and associated data structures [38].

Complex and increasingly dense UAV traffic adds a notable burden to the administration of air traffic, metropolitan plans and the assignment of resources. Given this context, a number of key questions are typically voiced, such as: Do we know if an appropriate route, in terms of airspace safety and energy efficiency for a specific mission, at a specific time, can be found ahead of time? Is it necessary to postpone the launch of some UAVs in order to accommodate a higher-priority mission, scheduled at a specific time? Responding to these queries and having the ability to predict traffic characteristics in advance, would facilitate more effectual preparation and regulation.

Moreover, one challenge to integrating UAV operations in the NAS involves capacity estimation. In other words, within a particular UAM airspace, how much traffic can be safely and effectively managed? In fact, UTM inherits the issue of capacity estimation from the ATM domain, where it has long been the focus of scholarly interest [39]–[41]. ATM primarily deals with pre-planned, airport-to-airport flights, however, while UTM implicates numerous users with less predictable demands, differing levels of experience, and the option of starting and finishing journeys almost anywhere. In other words, the non-deterministic component of small, unmanned-aerial-system (sUAS) traffic is a key difference between UTM and ATM.

While the potential advantages are immense, the deployment of AI tools and data science within the aerospace sector is not without difficulty. Above all, aerospace engineering must prioritize safety. Thus, any data-driven paradigms must generalize beyond training data. They must also be verifiable, certifiable and, from a human perspective, explainable and interpretable.

Ultimately, based on the current understanding of the latest conceptual and implemented frameworks, several gaps exist in the potential development of a data-analytic framework for the routine, safe, and efficient integration of UAV operations into civil airspace processes. The following are the main gaps that have been identified:

1. **Knowledge Discovery & Information Extraction**: Data mining helps extract beneficial information from enormous datasets; this is commonly

known as knowledge discovery. There is a lack of studies, deploying UAV trajectory data, which might use advanced data-analytics tools and technologies, with an aim to characterize air-traffic behavior for monitoring airspace, managing air-traffic flow, carrying out performance assessments, and improving airspace design. Although the ATM domain is rich in such studies, limited studies have been made in the UTM context. These techniques may help identify capacity indicators, extract safety models, identify risks, and provide decision support for better ATM systems.

2. **UTM Traffic-Flow Prediction (UTFP):** The thesis found that most of the existing literature either addresses the prediction of trajectory, or it deals with finding and resolving conflicts. Consequently, there is scant predictive research as to how air traffic will flow for UTM systems. With the information supplied by a congestion-prediction system, it will also be possible to plan a safe flight trajectory in advance.
3. **UTM Demand and Capacity management:** As conventional air-transportation demand constantly grows, air-traffic flow and capacity management (ATFCM), one of the three pillars of ATM (along with air-traffic services and airspace management), has been playing an increasingly important role in balancing traffic demand and airspace capacity. The airspace threshold will ultimately be reached as a consequence of the rapid increase in UAS-related demand: There may, therefore, be additional need for similar ATFCM initiatives.
4. **Lack of interpretability and explainability**: In air-traffic management, AI is receiving much attention, and it is already producing tangible results. Before AI/ML can be deployed more widely in UTM operations, however, important challenges must be overcome. There is as yet, for example, no commonly accepted framework for ML to be employed in the sphere of aviation. Moreover, the autonomous operation of unmanned aircraft systems involves safety-critical issues. Conversely, the traditional opacity of intelligent algorithms limits the usage of such systems in these areas, due to an absence of “explainable features” and knowledge

representation; such absences undermine the human ability to monitor, or even to comprehend proposed solutions. Thus, methodologies need to be developed with a view to making ML systems more transparent.

5. **UTM Traffic data (historic data):** There is a necessity to evaluate open-source datasets available for UTM, and to establish how these can be utilized in addressing core aspects of safely, enabling large-scale BVLOS UAS operations in urban environments. Not only is aerial dataset acquisition hazardous, however; it is also complicated by regulation, which presently restricts flights to thinly populated regions, and to conditions that fail to reflect the long-term objectives of UTM deployment. (In other words, autonomous use over areas of dense population.) If models are trained on data harvested under such unsuitable conditions, they will fail to generalize to the projected operational environment. In the literature, numerous scenarios have been proposed to resolve specific UAV scenarios, but they have either assumed a static environment, or they have not considered the effects of airspace-structural configurations or environmental constraints, such as weather conditions.

These lacunae were filled in Chapters 2-5, with each chapter focusing on a different objective. In the next section, a summary of the objectives is presented.

1.3 Aim and Objectives

In summary, the research aims to investigate the challenges and opportunities in the field of data analytics for effectively delivering essential UTM services. This will involve the development of a data-analytics framework capable of analyzing historical UTM data to predict air traffic flow patterns and generate recommendations to enhance airspace safety and capacity.

In order to bridge the aforementioned knowledge gaps and address the aim, a series of research **objectives** are set, as follows:

1. Conduct an in-depth literature review on the application of data analytics. Examine relevant research in ATM, UTM, and machine-learning techniques, and identify and analyze challenges and knowledge gaps concerning the improvement of airspace safety and capacity.

2. Design and simulate practical scenarios for drone operations, e.g., delivery services to generate UAV flight datasets, that take uncertainties into account, such as weather conditions, UTM emergency operations, and structural airspace configurations.
3. Develop a data-analytics framework for UTM applications, aiming to identify and characterize UAV trajectory patterns under variable airspace structures and environmental factors.
4. Develop a deep-learning model to predict air-traffic congestion, address issues related to UAV traffic management, enhance the mitigation of congestion, and promote efficiency in air-traffic operational systems.
5. Develop an explainable advisory system to support UTM demand and capacity-management services, by quantifying airspace availability and highlighting the most suitable regions for UAM operations.

1.4 Research Design

Figure 1-1 illustrates the connections between the thesis objectives, which comprise five phases. The methodology employed within each phase of this thesis addresses specific gaps relating to the use of data analytics within UTM management, with a view to enhancing airspace safety and capacity. Details about each of these phases are presented as follows:

Phase 1: This phase aims to identify research gaps and address the first objective. In this phase, literature will be reviewed comprehensively as it pertains to the application of data analytics and machine-learning techniques in UAV traffic-management systems. Popular and widely used search engines (such as Scopus, Google Scholar, and Science Direct) have been chosen to retrieve research articles for the current study. The material examined addresses (e.g.) the challenges of integrating UAVs or drones into the National Airspace System, and how Explainable AI and Data Analytics might contribute to a safer and more efficient airspace. In general, a review of the literature will facilitate comprehension, and the identification of various knowledge gaps. Thus, the overall subject context will be better understood.

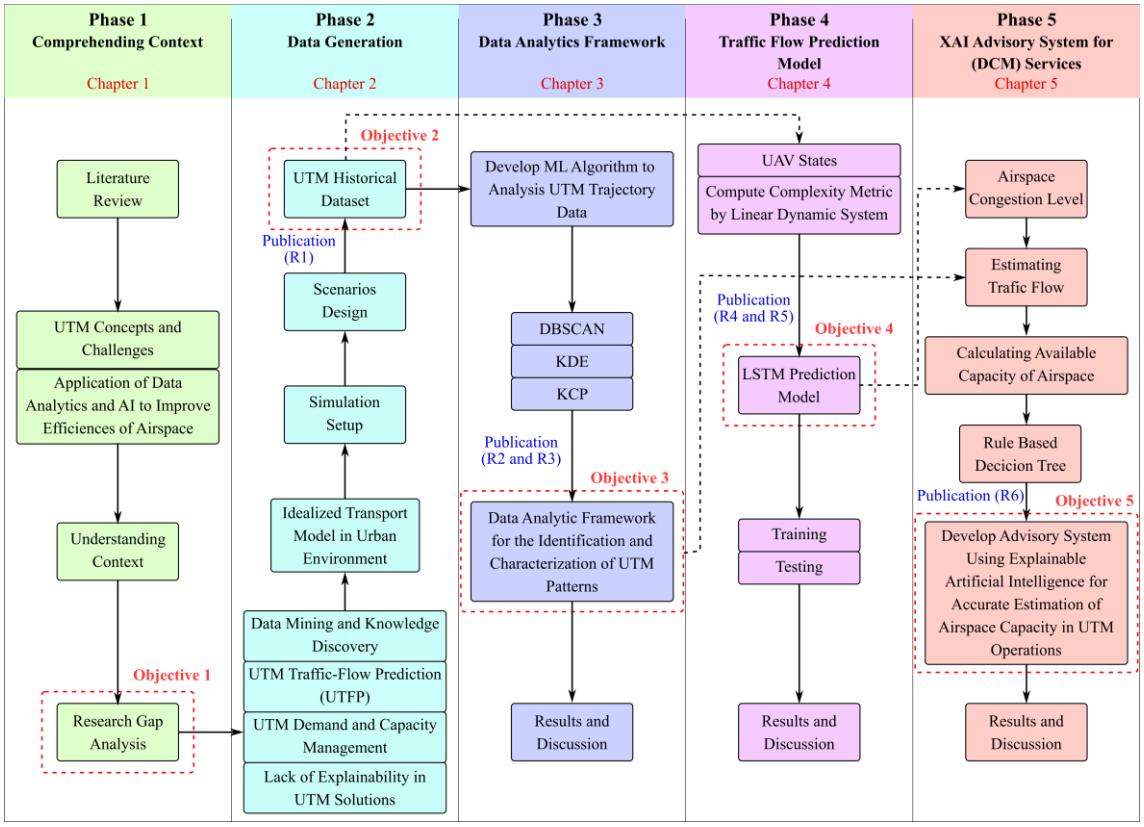

Figure 1-1 Thesis workflow diagram

Phase 2: The non-availability of historic UTM traffic data requires the generation of data via a simulation framework. In this simulation, a particle swarm optimization based (PSO) optimization algorithm is used to identify the best routes from a UAV service origin to its destination, even if there are (modeled) static and moving obstacles in the way. Validation of the proposed model was undertaken via a drone-delivery system. Scenarios included package delivery, emergency fire-surveillance operations, an inspection of the railway infrastructure with the use of UAVs, and the essential dispatch of Covid-19 test samples, with differing priority levels assigned to the operations in question. The simulation also addressed the potential impact of recreational drone use, at random times, during the execution of the tasks cited above. Moreover, the present research considers the influence of environmental factors (such as weather patterns), airfields with varying availability for UAV use, emergency UTM operations, recreational areas, and structural airspace configurations, such as static no-fly zones. During this phase, the first research gap has been filled, addressing the second objective.

Phase 3: The second gap was bridged during this phase, thus achieving the third objective. In this phase, a data-analytics framework is proposed that aims to characterize traffic-flow patterns in UTM airspace. The historical simulated trajectory data, gathered in the previous phase, is used to discover relevant knowledge from trajectory datasets on traffic abnormalities, UAV travel behavior, and UTM airspace patterns. As regards the identification of UAV traffic-flow patterns, clustering represents a commonly deployed technique for data analysis, particularly in the spheres of machine learning and statistics. Concepts of distance and similarity are central to the issues surrounding clustering, since the latter involves the grouping together of entities with similar characteristics. It is considered a first-tier trajectory-mining method, to determine heterogeneity and homogeneity based on data properties. Thus, similarities within a trajectory dataset are revealed by dividing the trajectories into various clusters. A number of clustering algorithms have been used to determine spatial and temporal traffic patterns, using flight data. The most popular cluster algorithms, such as k-means, DBSCAN (Density-Based Spatial Clustering of Applications with Noise), and OPTICS, are discussed in Chapter 3.

Phase 4: This phase describes the development of the deep-learning algorithm that will be used to predict air-traffic flow in the context of UTM, with the aim of alleviating the third research gap and addressing the fourth objective. A vast range of modeling techniques have been used in the development of air traffic-flow prediction (ATFP), and these can essentially be classified into four cohorts, namely, flight plan-based algorithms, methods based on statistics, and traditional methods of ML and DL algorithms. Chapter 4 discusses these methods and some related literature.

Phase 5: This phase aims at addressing the fifth objective, in order to fill the fourth and fifth research gaps. In this phase, suitable approaches were identified to implement demand-and-capacity-management (DCM) services in the UTM system. Such a system should have a range of incorporated functions (e.g., trajectory allocation, flight planning, and the optimization of airspace capacity). Moreover, in order to address the performance/explainability trade-off, various

explainability techniques have been considered. The literature presented in Chapter 5 has, in fact, identified methods of explainability. The first pertains to ML systems specifically designed with human interpretability in mind; these are so-called transparent ML models (e.g., logistic regression and decision trees). The second pertains to the explanation of models that are actually shallow; this is termed post-hoc explainability. Examples of post-hoc explainability approaches are local explanations, feature relevance, explanations by example, visualization, model simplification, and text explanations.

1.5 Thesis Structure

The core of this thesis is based on peer-reviewed journal publications. The complete list of publications related to thesis is shown below:

- **Publication (R1)**: Alharbi, A. Poujade, K. Malandrakis, I. Petrunin, D. Panagiotakopoulos and A. Tsourdos, "Rule-Based Conflict Management for Unmanned Traffic Management Scenarios," 2020 AIAA/IEEE 39th Digital Avionics Systems Conference (DASC), 2020, pp. 1-10.
- **Publication (R2)**: Alharbi, I. Petrunin and D. Panagiotakopoulos, "Identification and Characterization of Traffic Flow Patterns for UTM application," 2021 IEEE/AIAA 40th Digital Avionics Systems Conference (DASC), 2021, pp. 1-10.
- **Publication (R3):** Alharbi, I. Petrunin and D. Panagiotakopoulos, "Modeling and Characterization of Traffic Flow Patterns and Identification of Airspace Density for UTM Application," in IEEE Access, vol. 10, pp. 130110-130134, 2022, doi: 10.1109/ACCESS.2022.3228828.
- **Publication (R4):** Alharbi, I. Petrunin and D. Panagiotakopoulos, "Traffic Flow Prediction for UTM Application: A Deep Learning Approach," 2022 IEEE/AIAA 41st Digital Avionics Systems Conference (DASC), 2022, pp. 1-10, doi: 10.1109/DASC55683.2022.9925774.

- **Publication (R5):** Alharbi, I. Petrunin, and D. Panagiotakopoulos, “Deep Learning Architecture for UAV Traffic-Density Prediction,” Drones, vol. 7, no. 2, p. 78, Jan. 2023, doi: 10.3390/drones7020078.
- **Publication (R6):** Alharbi, I. Petrunin, and D. Panagiotakopoulos, ‘’Assuring Safe and Efficient Operation of UAV Using Explainable Machine Learning’, Submitted to MDPI Drones, manuscript id (drones-2317537).

This thesis consists of seven chapters. There are four technical chapters, as presented in the published papers. Details about each of these are provided below:

Chapter 1: The general research background and context are presented in this chapter.

Chapter 2: This chapter outlines the requirements for an idealized transport model, taking into account the regulations set by the Civil Aviation Authority (CAA), the UAS Traffic Management (UTM) Concept of Operations, and the SESAR U-Space. Moreover, this chapter gives an overview of the open-source software that can be used to provide UAV data collection and flight paths. The data-generation framework, weather-factor implementation, and deconfliction strategy are also described in this chapter which was originally published in paper R1.

Chapter 3: This chapter presents a new, proposed data-analytics framework to characterize traffic-flow patterns for UTM airspace, by analyzing simulated historical data. The details of this framework implementation are presented in the publications R2 and R3.

Chapter 4: This chapter presents the implementation of deep learning for traffic-flow prediction, in the context of UTM applications. In this study, air-traffic congestion will be addressed via a deep-learning methodology, across a timeframe of three minutes. The documentation for the proposed algorithm is provided in paper R4. Furthermore, the implementation of the deep-learning

algorithm is extended to include more analysis and discussion. The results are discussed and presented in paper R5, cited above.

Chapter 5: This chapter describes the implementation of the demand-and-capacity management advisory system for the UTM system. The implementation of this advisory system is presented in paper R6.

Chapter 6: This chapter presents a general discussion of the results. The chapter focuses on the findings of the research in terms of the real-world impact it will have.

Chapter 7: This chapter presents the general conclusions and makes suggestions for future research.

1.6 Contribution to Knowledge

The contribution of this thesis lies in the development of a data-analytics framework to optimize UTM network performance, while increasing the safety, efficiency, and capacity of airspace. The suggested model has been validated using simulated scenarios of UTM operations (e.g., drone delivery applications) that consider uncertainties due to weather conditions, static and dynamic obstacles, and emergency operations, especially in urban environments. The overall contributions of this research can be summarized as follows:

1. This study undertakes a comprehensive analysis with the objective of assessing the application of data analytics and AI tools in enhancing the safety and capacity of Unmanned Traffic Management (UTM) systems. The primary goal of this analysis is to conduct a thorough examination of existing UTM systems, gaining insights into the complexities inherent in future air transportation systems, and outlining the essential components for an urban air traffic management system. Furthermore, the study delves into the challenges and prospects associated with employing machine learning (ML) to deliver crucial UTM services effectively.

2. This thesis introduces a data-driven approach for the characterization of the UTM airspace structure and the operational performance of air traffic based on simulated historical data. Unsupervised learning is applied through flight trajectory clustering analysis to automatically identify spatial and temporal UAV trajectory patterns. Additionally, the UTM traffic flow is characterized through the application of three innovative quantitative metrics: Distance from the Centroid, Distance to Mission Completion, and Time to Mission Completion. These metrics are employed to evaluate deviations anomaly detection in UAV missions and their potential implications for UTM airspace safety and capacity.
3. The necessity for effective UAS Traffic Flow Management (UTFM) initiatives is essential to mitigate congestion and ensure safety and efficiency. To address this gap the present study proposed a learning-based model to predict air traffic congestion over a period of three minutes. The proposed model was adapted to make it suitable in terms of the look-ahead time horizon of the UTM applications (such as drone delivery services and emergency operations). With the information supplied by such a congestion prediction system, it will also be possible to plan a safe flight trajectory in advance.
4. This study enhances UTM capabilities by introducing a data-driven approach to identify and predict available urban airspace. The methodology centers around two critical components of Demand and Capacity Management system: predicting congestion levels for UAV trajectories and estimating airspace capacity. Through data analytics, congestion is categorized into five levels, and DBSCAN algorithms are utilized to determine the availability of urban airspace for Urban Air Mobility (UAM) operations.
5. This thesis enhances decision-support system transparency in the Unmanned Traffic Management Decision Support System (UTM DSS) by introducing eXplainable Artificial Intelligence (XAI). The study proposes a transparency-based methodology that combines Black-Box and explainable White-Box models. While Black-Box models lack transparency

in their internal configuration, White-Box models exhibit observable and understandable behaviors. The study introduces metrics-based scoring to assess the overall explainability of the hybrid model, based on the transparency of its individual components. Based on these metrics, the proposed advisory system demonstrates approximately 70% explainability.

2 UTM SIMULATION OVERVIEW AND IDEALISED TRANSPORT MODEL REQUIREMENTS

2.1 Introduction

To ensure that the UTM simulator can adapt to evolving UTM technologies and rules, and to address the lack of historical UTM data, it is necessary for the simulator to be flexible and expandable. This chapter presents a survey of available open-source software, simulation frameworks, and platforms that can aid in the development and testing of UTM functionalities. The incorporated survey compares their main features, while highlighting certain limitations of the open-source software tools that needed to be resolved to achieve a flexible and expandable UTM-simulator solution. Since path planning is one of the critical components of these simulators, this chapter also discusses the challenge of finding an optimal path between UAV sources and destinations, and it provides a retrospective view of various path-planning techniques.

Extensive research has been conducted to address performance issues in drone-based delivery systems. As the demands of UAM operations increase and environmental conditions become more dynamic, the design challenges for drone operations become more complex. This section introduces the requirements of an idealized transport model for UAV drone delivery missions, along with a Use Case Model based on UML notations. These requirements will later be translated into case study models in chapters 3, 4, and 5.

Furthermore, the chapter provides a detailed discussion of the simulation framework that was used in this study to generate data: this involved the development of various weather models, including factors such as rain, wind, and extreme weather fronts. This section also discusses the integration of different deconfliction strategies into the simulation tool, to resolve potential UAV conflicts.

2.2 Idealized Transport Model Requirements

The Urban Air Mobility (UAM) paradigm, jointly introduced by the Federal Aviation Administration (FAA) and the National Aeronautics and Space Administration (NASA) of the United States, envisages the aerial transportation of passengers

and cargo using novel electric air vehicles across diverse urban areas [42]. Recent advancements in automation, telecommunications, and electrochemical energy storage have paved the way for pioneering aerial transport solutions, which hold the potential to profoundly enhance urban mobility. Furthermore, the advent of drone-based delivery systems is emerging as a viable avenue for future e-commerce, logistics operations, and an essential component of smart city infrastructure. Esteemed entities, including Google, the Federal Aviation Administration, and Amazon, have embarked on experimental endeavors to ascertain the feasibility of utilizing drones for package delivery [43]. Nonetheless, the actualization of these urban aerial transport modalities encounters substantial impediments. The technological impetus driving these avant-garde aerial designs presents considerable quandaries pertaining to safety and security. Anticipating a surge in flight numbers, there's a pressing need for innovative air traffic control (ATC) frameworks for low-altitude operations to manage this augmented intricacy.

While the overarching premise of drone-based delivery systems appears to be advantageous, myriad factors can adversely influence their operational efficacy. Primarily, the energy constraints imposed by the limited battery capacities of drones circumscribe their flight durations, potentially inhibiting coverage of extensive areas. Additionally, environmental vicissitudes such as strong wind gusts, snow, or rain can deleteriously affect the performance of these aerial delivery mechanisms. Another pivotal consideration is cost-effectiveness. Service providers must ensure that delivery charges remain economically viable for consumers, lest the perceived exorbitance deter usage. Consequently, in the logistics domain, enterprises aspiring to assimilate drone-based delivery into their service offerings grapple with a plethora of challenges. These encompass delivery mission blueprinting, strategic drone routing and recharging, and the feasibility of multi-package deliveries within a singular mission. The intricate interplay of these dynamic elements amplifies the complexity of overarching planning [44].

This section outlines the requirements for an idealized transport model in accordance with CAA regulations [45] , UAS Traffic Management (UTM) Concept of Operations [46] ,SESAR U-Space [47] [48]. The core requirements, along with some underlying assumptions, are as follows:

1. Feasibility of Designing Optimal Trajectory

Designing optimal trajectories presents a significant challenge in the field of drone-based delivery systems, due to its crucial role in reducing overall transit times. Given the heterogeneity of delivery destinations and recharging locales, pinpointing a trajectory that simultaneously minimizes delivery intervals and ensures the drone's safe return to the depot proves arduous [49]. The salient criteria for optimal UAV path planning within this transport paradigm are delineated as follows [50]:

1) Optimality: This describes the time efficiency, cost efficiency and energy efficiency of the UAV system. Optimality can be categorized as non-optimal, suboptimal and optimal.
2) Completeness: This defines the criteria for finding an optimal path between UAVs. It allows a platform to find an optimal path via a suitable path-planning method.
3) Cost efficiency: This reflects the total computational cost of UAV communication over the network. Several factors affect cost efficiency, e.g., the cost between two nodes, battery-charging costs, memory costs, hardware and software costs, and fuel costs.
4) Time efficiency: This references the target operation of the UAV, from source to destination point, in the minimum possible time, while considering obstacles in the surrounding area. An optimal, short path, as taken by the UAV to its destination point, improves time efficiency.
5) Energy Efficiency: This defines the energy consumed by the UAV in terms of battery power, fuel, and energy required for performing target operations.
6) Robustness: This refers to the capability of UAV to tolerate specific device errors, linear driving errors, and rotation driving errors, while computing an optimal path.

7) Collision avoidance: This variable references UAV mechanisms that consume power to detect obstacles without any physical damage.

2. Drone Types and Energy Consumption Model

Energy consumption stands as a paramount metric for drones, a significance underpinned by their restricted payload allowance, finite battery reserves, and circumscribed flight durations [51]. To facilitate an accurate projection of drone flight duration and battery status, a rigorously precise mathematical model delineating energy consumption is essential [52]. This model ought to incorporate variables including the drone's mechanical design (e.g., rotor count) relative to its classification, as well as external factors such as wind dynamics, gravitational effects, achievable velocities, traversed trajectories, operational altitudes, and extant payloads [53].

3. Considerations of Delivery Points

In the context of drone-based delivery missions, the drone can be assigned to either single or multiple destinations. For instance, within UTM airspace, missions are typically categorized as either linear trajectory missions or area missions [48]. The distinction between these two types of missions becomes evident when examining the temporal and spatial characteristics of their trajectories. In area missions, certain positions may be visited multiple times by the unmanned aircraft system (UAS), whereas in linear missions, each position is traversed only once. In practical terms, this means that the drone must collect a parcel from a specific location or depot and, after delivering the parcel, it must determine whether to wait for the next parcel or return to the depot. These decisions depend on factors such as the drone's current energy level and the delivery route. Hence, it is of utmost importance to develop optimal routing decision-making algorithms that take these conditions into account.

4. Payload Consideration

An increased payload weight has a negative impact on the performance of drone-based delivery systems. This is primarily due to the battery constituting a major portion of the drone's weight, in addition to the weight of the parcel itself. As the weight increases, the flight time of the drone decreases significantly. Therefore,

it is important to take payload constraints into account when designing efficient and dependable drone-based delivery systems. Research studies that consider both payload and battery weight have demonstrated notable reductions in overall costs and flight time in the context of drone-based delivery [54].

5. Priority-Aware Trajectory Design

Due to the urgency of some flights, it is necessary to prioritize these over less pressing services. For instance, this is relevant for emergency service flights, including air ambulances or search and rescue services, which require the ability to move freely in airspace and have fast, effective responses when involved in conflicts. In light of this, NASA UTM proposed, in its “prioritization” recommendation, that each aircraft service receives a varied priority level [55]. Specifically, the U-space recommendation wished to introduce eight priority levels, numbered between one and eight, where eight is the lowest priority [56].

6. Weather Conditions

Considering weather conditions is essential to enable drone-based delivery operations during emergency situations. Adverse weather conditions, including strong winds and heavy rainfall, can have a substantial impact on drone flights. As a result, reliable weather data serves as a critical prerequisite to ensure the dependable performance of drones. For instance, in cases of significantly high wind speeds or increased rainfall, it is advisable to suspend the flight to prevent potential damage to the drone, which would subsequently incur additional costs. Therefore, based on periodic weather data, such as wind and precipitation levels, the system can make informed decisions regarding whether to proceed with the mission or recall the drone to the depot [57].

7. Uncertainty-Aware Delivery Service

In the realm of drone-based delivery services, various uncertain factors come into play, including weather conditions, fluctuating payload weights, drone airspeed, and customer demand [58], [59]. Neglecting these subtle factors leads to increased costs and flight times, thereby compromising the overall performance of the delivery service. Uncertain conditions can also contribute to drone malfunctions, which can have cascading effects on subsequent parcel deliveries.

In such cases, the penalties incurred are significantly higher, further driving up the overall costs [60]. Moreover, critical safety considerations include the presence of no-fly zones, maximum allowable flight altitudes, nighttime operations, and maximum load capacities, all of which must be taken into account.

The omission of these uncertain conditions in problem formulation is a prevalent limitation in most existing studies, presenting an ongoing challenge [44]. Consequently, it is crucial to place substantial emphasis on addressing these factors to ensure that the developed approach can effectively handle such real-world scenarios.

8. Collision Avoidance & Separation Management

Despite the recent introduction of single drone-based delivery systems, the use of drone swarms has become necessary to meet the growing demands of deliveries. Consequently, it is essential to incorporate effective collision avoidance techniques to ensure that drones within the swarm do not collide during their flight operations. Moreover, urban areas pose additional challenges, with numerous high-rise buildings and other obstacles creating a highly dynamic environment. In general, an intrusion occurs when there is a loss of minimum separation between two aircraft. While a loss of separation (LOS) does not always indicate an impending collision, it signifies that two aircraft are closer than the accepted safety distance. Therefore, conflict avoidance systems aim to prevent one aircraft from entering the minimum safe separation zone of another aircraft. When a loss of separation is predicted to occur in the future, it is referred to as a conflict. Once a conflict is detected, the aircraft is redirected along a deconflicting path. Designing an effective collision avoidance algorithm requires consideration of various factors, including data transmission latency, computation latency, and control response latency [57].

In the context mentioned above, an optimal transportation model necessitates the establishment of effective horizontal and vertical separations between unmanned aerial vehicles (UAVs). However, many collision detection and resolution (CD&R) studies rely on the International Civil Aviation Organization's (ICAO) definition,

which mandates a horizontal separation of 5 nautical miles (NM) and a vertical separation of 1000 feet [61]. In the realm of unmanned aviation, no pre-defined standard separation distance exists. Nonetheless, a commonly utilized value in research is 50 meters [62].

9. Deconfliction Strategies

A conflict is defined as the predicted loss of both horizontal and vertical separation between two or more aerial vehicles [63]. It is necessary to develop an efficient conflict management system that must be able to detect conflicting traffic in sufficient time and perform avoidance maneuvers to resolve conflicts [64]. ICAO elaborate the deconfliction in more details and address the need for strategic, pre-tactical and tactical deconfliction methodologies to be applied as appropriate [65].

10. Partitioning and Sectorization of Airspace

Another important consideration in UTM transport model is the partitioning or sectorization of the airspace to ensure safe and efficient air traffic flow [66] . Different methods have been proposed to partition airspace, such as the Improved Binary Space Partition Method [67] ,Voronoi-based (convex polygons) or simple grid-based partitioning [68].

11. Regulations

The utilization of drones in logistics represents a novel approach in the commercial drone industry. However, this emerging field faces various challenges and tasks that must be addressed to enable the successful operation of autonomous UAVs. For instance, companies like Amazon have encountered regulatory hurdles in their pursuit of commercial drone use. The Federal Aviation Administration (FAA), the governing body responsible for air traffic regulations, only granted approval for the commercial use of Amazon's drones in 2022 [69]. According to Sah et al. [70], challenges related to regulations, as well as concerns regarding privacy, security, public perception, environmental impact, technical aspects, and economic factors are among the primary obstacles to the implementation of drones in the logistics sector. Similarly, Çıkmak et al. [71] have identified regulatory issues as the foremost challenge, followed by security and

safety concerns. Therefore, it is crucial to ensure that drones used for delivery purposes adhere to relevant rules and regulations, as compliance with these guidelines is an integral aspect to be considered.

2.2.1 Idealized Transport Model Use Case Diagram

In the UAS Traffic Management (UTM) Concept of Operations [46], the UTM is defined as a collaborative system that relies on community-based cooperation. Under this framework, operators and entities providing operational support services have the responsibility of coordinating, executing, and managing operations while adhering to the established rules of the road set by the FAA. Commercial UTM Service Providers (UTMSPs) will play a crucial role in delivering various UTM services aimed at managing traffic for small Unmanned Aircraft Systems (sUAS). These providers will offer a range of services to sUAS operators, including flight planning, communications, and separation assistance [72]. In order to demonstrate how the idealized transport model requirements work together cohesively, as discussed in the above section, the Unified Modeling Language (UML) [73] use case modeling approach is employed.

The use case diagram captures a system's functional requirements from the users' perspective. It provides a high-level representation of the available functionality and the interaction between actors. The diagram also depicts the associations between services/functionalities and actors. The major service provider is represented as a module, and a component is defined as a group of modules in UML design notations. UML is an extensible language, allowing the introduction of new elements for specific domains if needed. Stereotypes, such as <<manages>>, <<informs>>, <<plans>>, and <<updates>>, in addition to conventional <<includes>> and <<uses>> stereotypes, are used to illustrate the idealized transport model requirements. These models are depicted in Figure 2-1 below:

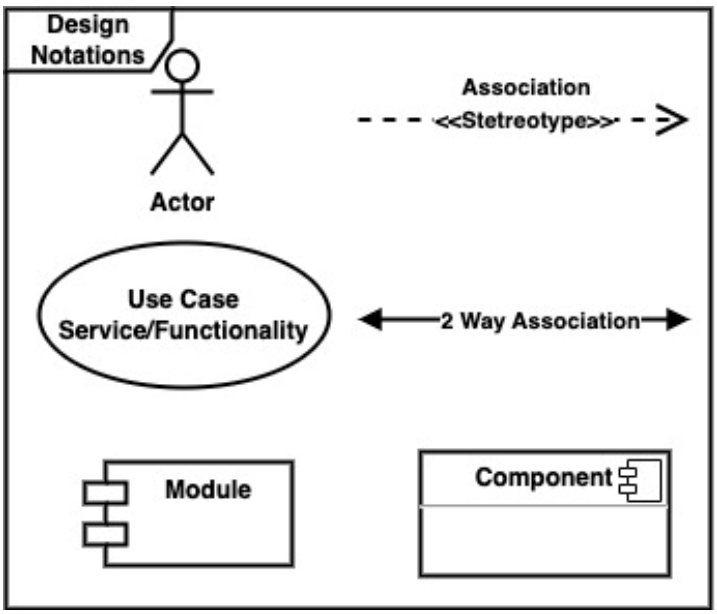

Figure 2-1 Use Case Model Design Semantics [73]

The idealized transport use case model is presented in Figure 2-2 , which specifies two main actors in the system: the UAV operator and the UTMSPs.

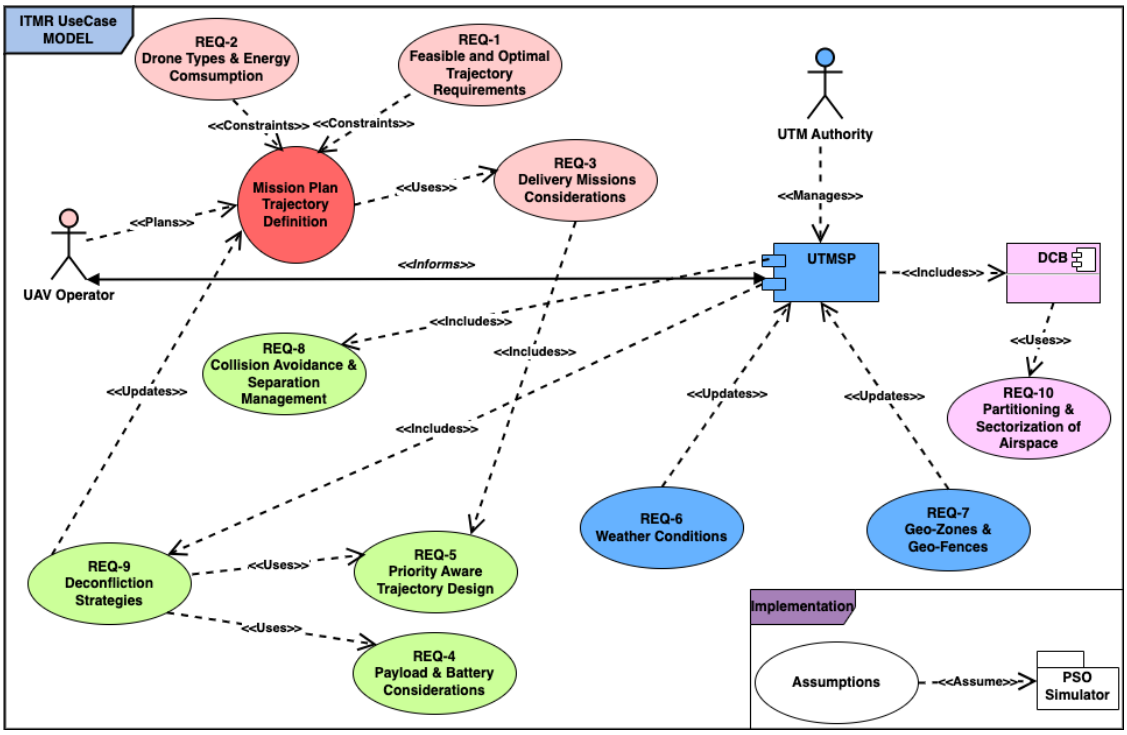

Figure 2-2 Idealized Transport Model Use Case

UAV Operator:

1. ***UAV Operator*** *plans the mission using the* ***Mission Plan service/functionality.***
2. *There are two constraints that are to be considered in defining a trajectory as specified in our model as two cases* ***Trajectory Feasibility*** and ***Drones Types & Energy Consumption***.
3. *The* ***Mission Plan <<Uses>> Delivery Missions Consideration*** *that* ***<<Includes>> Priority Aware Trajectory Design*** *taking in account priorities assigned to each mission.*
4. ***UTM Authority <<Manages>> UTMSP*** *Module that* ***<<Includes>> Deconfliction Strategies*** *service. As per* ***<<updates>>*** *from* ***Deconfliction Strategies*** *service, the* ***Mission Plan*** *is tailored as required.*
5. ***UTM Operator <<informs>>*** *about any critical event and data to* ***UTMSP****. It also receives any recommendations or data from* ***UTMSP*** *to conduct safe and efficient UAV operations. This is illustrated using double arrow association symbol.*

UTM Authority:

6. ***UTM Authority <<Manages>> UTMSP*** *Module.* ***UTMSP*** *also* ***<<Includes>> Deconfliction Strategies*** *service that assists in conflict management.*
7. ***UTMSP*** *also* ***<<Includes>> Collison Avoidance & Separation Management*** *service to enhance safety of airspace.*
8. ***Weather fronts*** *&* ***Geo-Zones*** *or* ***Geo-Fences*** *service* ***<<updates>> UTMSP*** *regarding weather events and dynamic airspace configurations.*
9. ***UTMSP*** *Module* ***<<includes>>*** *Demand and Capacity Balancing (****DCB****) Component.*
10. ***DCB*** *Component* ***<<Uses>> Partitioning & Sectorization of airspace*** *for managing airspace demand and capacity balancing requirement.*

2.2.2 Assumptions and Considerations for Case Study Implementation

This section outlines the assumptions and considerations taken to simplify and simulate the delivery scenarios within the case study transport models. The underlying assumptions are detailed below:

1) The scope of this study assumed that the bandwidth is sufficient to meet the UAVs' rate requirements, and interference is trivial since different users and UAVs use different frequencies. This assumption aligns with the guidelines provided by the Civil Aviation Authority (CAA) regulations [45] and the Specific Operations Risk Assessment (SORA) methodology [74]. By ensuring reliable communication between the Ground Control Station (GCS), operators and relevant authorities,

drone operators can effectively guarantee the safe execution of complex operations.

2) The SORA methodology requires that batteries are fully charged and securely mounted [75]. The missions under consideration do not include the aspect of battery swapping or charging through intermediate depots. It is assumed that a battery will support a drone for one mission, and upon returning to the depot, the battery will need to be changed or recharged.
3) For UAVs delivery service scenarios, it is assumed that each location's demand will be completely fulfilled by a single drone. Upon the drone's arrival, the customer will promptly receive the goods, and the weight of the delivery package will not surpass the drone's payload capacity.
4) According to the European Union Aviation Safety Agency (EASA) [76] , the newly established regulatory framework encompasses three distinct categories of unmanned aircraft systems (UAS) flights. The first category, known as the Open category, encompasses operations with low levels of risk. In this category, operators are obligated to adhere to a limited set of operational rules, such as maintaining a safe distance from people. Additionally, compliance with safety requirements and adherence to specified mass limitations are also mandated. The second category, referred to as the Specific category, encompasses operations with a moderate level of risk, necessitating the acquisition of authorization from a national aviation authority. Lastly, the Certified category encompasses operations with higher levels of risk, comparable to those encountered in manned aviation. Consequently, the authorization requirements for the Certified category are notably similar to those applicable to manned aviation operations. In this study, two types of categories are considered: delivery operations scenarios using UAVs in urban environments are categorized as medium risk in the specific category, while recreational UAV users fall into the open category. This study assumes that the process of Specific Operations Risk Assessment (SORA) entails the comprehensive evaluation of

both ground risk, denoting the potential harm posed to individuals located on the ground, as well as air risk, indicating the potential danger to individuals within the airspace, throughout the delivery procedures. Furthermore it is assumed that the recreational user flight is analogous to ICAO's regulations [77] , by EASA [76] and the regulations for unmanned aircraft operations in the UK [45]:

- Each aircraft must be marked with the Operator ID for the owner.
- Prior permission is necessary to fly in or near airports and controlled airspace.
- Drones must not be flown above 400 feet (120 meters).
- Flying near people, congested areas, or sensitive locations such as prisons and military bases is also prohibited.

5) The flight range, payload capacity, and speed of each drone type are predetermined and established in advance.
6) To avoid complexity in the case study transport models, only the position (x, y) trajectory coordinates are considered, assuming UAVs fly at fixed altitudes and constant speeds.
7) In this study it is assumed that drones travel between locations at a consistent speed v in meters per second (m/s). This assumption is based on the notion that the energy expended by a UAV during flight is approximately equivalent to the energy expended during hover. However, it is worth noting that the energy consumed during flight might actually be slightly lower than the energy consumed during hover, potentially due to the phenomenon of translational lift.
8) Taking into consideration the ongoing discussions surrounding the vertical structure of airspace [78][79], the assumption of a constant altitude for drone flights, after vertically ascending at the origin and prior to descending at the destination, is reasonable in certain contexts. This simplification allows the focus of the study to be directed towards the significant aspect of utilizing data analytics. By assuming a constant altitude, the study can concentrate on the key elements, such as the

efficiency of a drone delivery system, horizontal traffic flow, and 2D capacity planning, within the UTM airspace.

9) Time and energy efficiencies are considered as metrics to measure trajectory efficiencies [80]. Additionally, the PSO-based trajectory generation algorithm always seeks the shortest path as one of its constraints.
10) Square or rectangular loiter-type area missions were considered for surveillance purposes. Various closed-loop polygon shape missions were also considered for delivery services. Additionally, the case study transport model takes into account the possibility of executing some services in a round-robin fashion according to a schedule.
11) NFZ (No-Fly Zone) and Dynamic Obstacles have been modeled as circular objects to address the geo-zones and geo-fencing requirements of CAA (Civil Aviation Authority).
12) In the case study transport models, two categories of weather were considered based on the Beaufort wind scale: adverse weather (wind speeds between 13-23 m/s) and extreme weather (wind speeds exceeding 33 m/s). The effects of adverse weather were modeled as turbulences in the trajectory path, while severe weather fronts were modeled as circular objects moving along the wind direction. The path planning algorithm incorporates these weather fronts as constraints and aims to plan trajectories away from these hazardous regions.
13) To design priority-aware trajectories in the case study models, the priority level suggested by SESAR U-Space [47] was followed.
14) In the case study model, the battery consumption was measured by calculating the total flight time between a start and end destination. In this study, the maximum rated battery endurance is half an hour.
15) The effect of weather conditions on battery consumption was left out of the scope at this stage. It is assumed that the UAV operator operates within desirable temperature ranges, and additional measures such as thermal management, including insulation or cooling, have been implemented in the UAV avionics section to mitigate the effects of

temperature on battery performance. Although wind conditions can affect the battery consumption of a UAV, the impact of wind on the battery itself is not considered in this study. It is assumed that the batteries are fully charged and capable of supporting the drone for a single mission.

16) In this study, certain assumptions have been made to facilitate the application of machine learning techniques and to manage computational complexity. One of the assumptions is that a maximum of 200 UAV flights will be considered. This limit has been set to strike a balance between obtaining sufficient data for analysis and ensuring manageable computational resources. Additionally, a safe separation distance of 50 meters between adjacent UAVs has been assumed. This value is based on industry standards and regulations, which prioritize safety and minimize the risk of collisions between drones.

17) Given the lack of historical UTM data, the present study proposes use cases that take into account airspace structure, emergency UTM operation, and environmental factors. This model considers the weather conditions of Bedfordshire, a region covering 64 x 64 km, which encompasses nine distinct areas where flights may encounter restrictions, including airfields, recreational areas, and a prison.:

    - Four airfields: Luton, Cranfield, Halton, and Old Warren.
    - Four recreational areas: Dunstable, Sandy, Cardington, and Graveley.
    - The inclusion of Milton Keynes Prison.

    By simulating this extensive region of Bedfordshire and incorporating the aforementioned aspects into the model, the aim is to provide a more accurate reflection of the real-world conditions in which UTM systems operate. This approach of creating a self-contained island, rather than sampling from a larger map, contributes to a more comprehensive and applicable model.

2.3 Overview of Simulation-based Tools for The UTM System

Simulation is a significant and cost-efficient tool, particularly when real-world implementation is linked to precarious conditions and considerable expense. Furthermore, simulators have other merits: with the help of simulators, researchers can experiment with different scientific topologies in limited timeframes, and with few resources. Research works associated with UAVs, including those addressing traffic management, represent a relatively new research field with (thus far) limited real-world implementation. Hence, simulation is the main avenue for the development of UTM, and for testing its functionality [81].

Moreover, most of the existing UAV conflict-detection and resolution methodologies are based on simulation and sensor control, because methods based on analytics require prior data (e.g., substantial, historic UAV trajectory and navigation data) to become operational. If prior data are not available, then the opportunity to apply data analytics is restricted. Nonetheless, a considerable barrier to data analysis in UTM is the absence of a common and shared database containing real-time UAV flight-operation data. One possible developmental direction is the use of simulation data [82]. In this scenario, researchers can virtually simulate a full UAV management system (before its real-world hardware implementation) with the help of controlled simulation-based experimentation. Limited resources comprise a critical constraint in the development of a real UTM system. Still, it is better to build the latter as soon as possible, in order to generate, and accommodate, a rapid increase in international research activities [83].

In order to simulate distinct applications and scenarios using any kind of UAV, a user-friendly and standard simulator with different characteristics is urgently needed. The newly designed simulator should work in such a way that it accommodates different algorithms, such as communication protocoling, obstacle avoidance, path planning, and indeed, much more [84].

Two distinct approaches may be used to develop a UTM simulator. One involves reliance on open platforms and sources, while the other entails proprietary solutions. Certain private companies, such as Airbus and OneSky, have

advanced their own, proprietary simulators. A similar approach has been adopted with the U-Space and NASA-UTM initiatives, cited earlier, since these organizations have long experience of working with ATM. The open-source approach, conversely, is typically favored by academics and smaller research institutions. Without the need to construct specialist tools and simulator platforms from scratch, such actors can address UTM-oriented research while avoiding excessive financial or labor costs [72], [85]–[88].

At present, various commercial applications such as Unifly [89] and Airmap [90] are operational, and these implement a majority of pre-flight services, while in-flight services are limited. Currently, furthermore, numerous in-flight services are either not completely built, or only partially designed.

One simulation-based study is that of Razzaq et al. [84]. The authors proposed a 3-D graph, theory-based routing algorithm, accounting for the de-confliction of autonomous aerial vehicles with obstacles in their way. In another study, Sedov and Valentin [91] studied strategies of de-confliction among simulated UAVs in high-density VLL uncontrolled airspace. This study comparatively simulated both centralized and distributed de-confliction algorithms for a “busy day”, consisting of 12 hours of UAV traffic. In another relevant work, Ho et al. [92] introduced a simulation-based de-confliction technique called Adapted ORCA. This proposal addresses real-life considerations that are immanent to the deployment of UAVs in shared airspace.

The researchers in [93] illustrated an easily distributable, economic simulator, which concentrated on simulating missions carried out by several UAVs, while permitting simultaneous extraction. This particular simulator, named Drone Watch and Rescue (DWR), focuses on simulating missions by considering several UAVs. Originating from the video-gaming field, DWR was designed with a two-level architecture, in which updated web-development technologies were used. The limitation of using such updated web technologies to carry out UAV simulations is, however, that they require high-performance systems, in which modern JavaScript engines evince efficient performance while computing intensive tasks.

In 2017, Microsoft designed an Aerial Informatics and Robotics Platform (AirSim). The purpose of developing such a model was to support the testing and development of algorithms for autonomous vehicle applications, such as reinforcement learning, computer vision and deep-learning algorithms [94]. It was developed in such an extended way that it could easily accommodate distinct kinds of software protocols, hardware platforms and other vehicles. For fixed-wing vehicles, nonetheless, AirSim only simulates thermal and advanced wind-effect simulations, whereas it does not simulate advanced ground-interaction and richer collision-response models [85].

Gazebo is an open-source robot simulation tool with different models and settings, including a model for the simulation of UAVs. The tool provides capabilities accurately and efficiently to simulate populations of robots in complex outdoor and indoor environments. For a number of technology challenges and competitions, Gazebo has been used as the simulation environment. An open-source framework is used for the design and development of modular robot applications, while UAV objects are simulated by using the tool in conjunction with Robotics Operating System (ROS). Gazebo is, in fact, one of the most famous platforms. A host of different sensor models, including a physics engine and the ability to generate 3D virtual worlds, are features that make it distinguishable from other simulation platforms. Additionally, various kinds of robot, such as a manipulative arm, are associated with Gazebo. This popular simulator, nonetheless, has certain limitations, in areas such as its legacy physics engine and photorealism [95].

As regards developing and testing the functionalities of a UTM simulation framework, an example has been given in [96]. The researchers in this paper used a simulator based on U-space services and functional architecture. The former are implemented as ROS nodes, whereas the interactions and interfaces of these nodes are depicted in full detail. Furthermore, Gazebo is used here to configure different customized scenarios that can easily combine rotary-wing and fixed-wing UAVs.

The Unity platform [97] was used to build a UTSim simulator. This type of simulator has certain capabilities for simulating UAV physical navigation, communication, specification, avoidance, control and sensing, in various environs with moving and static objects. Moreover, this simulator helps researchers and observers explore and study various issues regarding unmanned aerial-vehicle air-traffic integration, such as sensing and avoidance, navigation algorithms, communication protocols, and others [85]. The performance evaluation for UTSim is not available, however, because this particular simulation does not exist as an open source.

To carry out analysis and research work regarding air traffic, a Python-based tool known as “BlueSky” [98] was designed. The aim in designing this particular tool was to test the developments evinced by air traffic, along with certain characteristics such as simple interfacing location, and air-traffic flow with a satellite-view map. To permit expansion and customization, an open-source software was used. As it was developed for carrying commercial ATM/ATC, nevertheless, BlueSky does possess certain limitations within its simulation environment. Primarily, it solely offers services or simulations that are centered around modules designed for commercial-sized aircraft. Furthermore, the operator interface for modifying or configuring UAS flight trajectories is complex in nature. Additionally, the incorporation of real-time physics-based interactions is currently absent within the simulator [99], [100].

An air-traffic control tool named Openscope [101], meanwhile, is an advanced online simulator. This unique simulator incorporates several airports, plus an interface that can optimize departures and arrivals, and there is a command line that can easily modify aircraft trajectories. Nevertheless, this simulator also evinces certain drawbacks, outlined below.

a) It primarily focuses on analyzing the departures and arrivals of aircrafts at particular airports. Hence, the aircraft must strictly follow their established routes.

b) It is impossible to expand and customize the aircraft models in this simulator, because they are not open source.
c) This simulator does not aim to configure and create various scenarios with aircraft, as well with their interactions.
d) The models of aircraft flying in VLL airspace evince larger sizes than typical rotary-wing and fixed-wing UAVs.

The eUTM simulator is discussed in detail in [83]. Its principal purpose is to afford researchers a platform, via which they may interrogate UAV deconfliction algorithms for BVLOS operations at very low altitudes (where civil UAV flights are permitted). The eUTM simulator prototype can support several UTM services to the point of U-space or NASA's UTM. In turn, this facilitates the study of regulations, pertinent policies, deconfliction algorithms, etc. Nonetheless, various limitations must be considered, so that the eUTM simulator can be enhanced in terms of communication failure, boundary management and dynamic airspace configuration (e.g., dynamic geofences). Ebeid *et al.* [102] presented a survey of a range of publicly available, open-source UAV hardware and software platforms, which may be used for research and development related to drone simulation. Their study covers more than 20 flight controller platforms, which are publicly available on the web. Based on this survey and review of existing UTM/UAV simulation platforms and tools, two programming languages, named C++ and Python, stand out as the most popular simulator-development programming platforms. The aforementioned survey highlights both current advances and recent challenges. Further, this research work focuses particularly on specific drone simulator applications. The study also found, that although there has been significant development in building UAV simulators, the progress is limited to narrow domains of interest, such as training and the design of individual UAVs.

In general, most of these tools comprise commercially closed-source software, such as Airmap, Unify, and UTM-One Sky. The limitations regarding these tools are that accuracy of simulation, public information on models, and the development of APIs for autonomous applications are not considered. Therefore, they are impossible to expand or customize. Some researchers have developed

their own simulation environs, for their own custom-built UAVs and applications. Nonetheless, these are not well documented, and their performative comparison is not available, as illustrated in [103].

To conclude this section, the requirements for simulator design are presented in Figure 2-3 below:

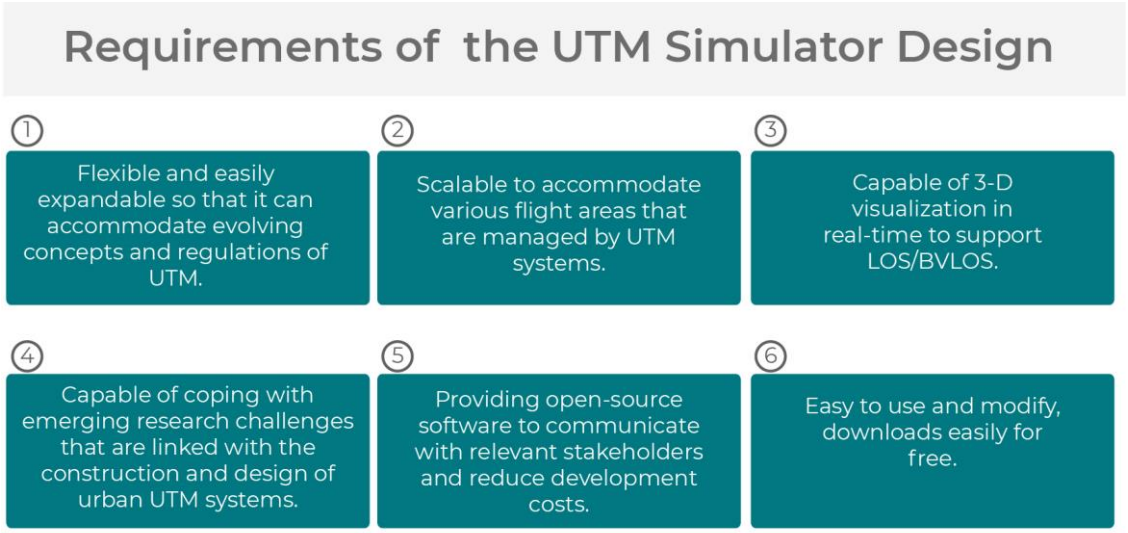

Figure 2-3 The requirements of the UTM Simulator

2.4 Path Planning Optimization

With its focus on identifying an optimal route between the origin and destination of a UAV, path planning is currently a critical field of research. In a dynamic environment, UAVs may encounter various obstacles during their journeys, and paths must be configured to prevent collisions. Three key terms pertain to UAV-oriented path planning, namely, trajectory planning, motion planning, and navigation [104].

Several research approaches have been proposed in recent years that describe the path planning of UAVs and address different issues. Researchers have focused on different path planning issues of UAVs, in the context of the complexity and shape of these vehicles. For instance, the authors of [105], [106] addressed different regions in terms of sweep directions for determining optimal UAV paths. Similarly, Torres *et al.* [107] provided forward and backward patterns for minimizing the distance between sub areas. The authors of [108] proposed

considering C space for the position vector of UAVs. Acevedo *et al.* [109], meanwhile, proposed spiral patterns for the path planning of UAVs in complex coverage areas.

Path-planning methods, deployed to find optimal UAV paths, tend to be computationally inexpensive in terms of time and money. These methods use the minimum time and energy necessary to provide a collision-free environment for UAV operations. In addition, these methods should be designed as robust and complete.

2.4.1 Path Planning Techniques

A brief literature review is presented here, with the most relevant studies in the field of UAV path planning algorithms. Followed by qualitative comparisons of the performances of these algorithms, are tabulated Table 2-1.

1. Particle Swarm Optimization (PSO)

In 1995, the technique of particle swarm optimization (PSO) was first pioneered by James Kennedy and Russell Eberhart [110]. The PSO algorithm is a heuristic optimization technology: it is used globally and founded on the stochastic population. This algorithm is particularly useful for robust implementation and swift convergence [111], and this makes the technology a popular choice for optimization issues and/or control systems. To identify the best probable solutions that correspond to optimization challenges, this technique involves persistent movement of the particles within the search process. There are three derivatives of the algorithmic function:

a) Optimization: The movement and velocity of each particle are collectively referred to as inertia. Every particle demonstrates unique direction and velocity, along with memory of the best fit solution for the target goal. The movement of the particle is contingent upon the swarm intelligence, best solution candidate, and velocity, when the optimal swarm position is achieved.
b) Collaboration: The information about best solution is exchanged between the particle and the swarm, which is followed by the consideration of an optimal solution by the swarm to the defined objective.

c) Adjustment: The adjustment of each particle is dependent upon its movement in three vectors. The random scalar time for each particle is noted when it moves between the local position and the optimal position of swarm, while utilizing the particular inertia. The particle illustrates this movement repeatedly, while coinciding with the number of simulations running in parallel. The particle comes closer to the global best solution after every movement.

In addition, PSO can be leveraged to count and determine the optimal values of trajectories from A to B. In so doing, PSO identifies the best possible trajectory, which is cost effective and efficient against any challenges. Since the PSO algorithm was used for anticipating and planning drone paths, it was particularly helpful in identifying the solution that requires the minimum movement between starting and end points, while avoiding any obstacles.

2. Probabilistic Road Map (PRM)

An essentially simple method, a probabilistic route map comprises a network of interconnected nodes. This method provides an obstacle-free path from source to destination point, using information about occupied and free space on the path [112]. It involves using graph searching methods, after constructing the road map from source to destination point. The route map starts with an initial configuration setting, before expanding randomly with node and edges [113].

The road-map method has been used to determine UAV path-related queries in a static environment. It can resolve UAV path-planning queries during the processing time deployed in real time for UAV path planning. Road-map methods have been used for smoothing the UAV path. For instance, Jang *et al.* [114] applied the road-map method to determine optimal UAV path-planning control. The authors also used these methods to interrogate a “travelling-salesman problem”. They applied 20 methods to calculate flight time by covering all remote sensing regions, while executing all UAV communication tasks.

3. Potential Field Method

A light, simple approach to dynamic path planning, the potential field method, or PFM, represents environments in order to facilitate the design of objects as particles [115]. Movement occurs under the control of potential fields, situated around the c-space. The resulting fields provide the basis for calculating UAV paths, from starting to destination point. Nonetheless, local minima, which cause "sticking" of the UAV before it arrives at its target, are problematic for the standard PFM. Resolution of the unreachable-destination issue via a hierarchical PFM was, therefore, suggested by Dai *et al.* [116]. Here, the authors sought to decrease local minima, and collision likelihood between UAVs, by the addition of rotational forces between vehicles. Meanwhile, Bai *et al.* [117] thought to improve multi-UAV PFM by introducing a PFM longitudinal factor, which had the additional benefit of resolving the issue of local minima between UAVs. In order to satisfy UAV performance requirements, these authors presented the B-spline method of interpolation.

There are many ways to overcome the shortcomings of PFM, and to avoid collisions in the planning of UAV paths, maintained the authors of [118]. A technique to reduce UAV collision likelihood was, for example, proposed in [119]. These authors presented a PFM model that was more time efficient than its conventional counterpart, and this new model could be deployed in UAV navigation. The enhanced PFM, known as APF, could be used both in path design *and* to attract UAVs for the requisite goal configurations. UAVs would, indeed, be repelled from both c-space obstacles and collisions in general. In the context of UAV path planning, this model became popular for applications designed to avoid obstacles and reduce collisions.

4. Genetic Algorithm

The genetic algorithm is one of the most commonly used metaheuristic search methods for interrogating efficient robot path planning. A genetic algorithm is used as a search algorithm. These algorithms are based on the evolutionary phenomenon, since the method imitates the natural process of selection, and survival of the fittest. Many researchers have focused on applying a genetic

algorithm to solve complex optimization problems in different domains, such as intrusion detection. The UAV path is encoded as a genetic-algorithm chromosome, which evolves during the algorithm [120]. The algorithm attempts to optimize a set of solutions as a fitness path, optimizing an objective function known as the fitness function. This function determines the fitness of each solution generated during the genetic-algorithm execution. Here, certain operators, such as mutation and crossover operators, are used, affecting the genetic-algorithm performance. Meanwhile, the genetic algorithm has the advantage of independence of computation time over the number of constraints [121].

From the comparison in Table 2-1, while Genetic Algorithms evince impressive convergence speeds and success rates for fixed points, they easily converge to local rather than global minimums. Compared with GA, PSO requires fewer parameters to be adjusted for algorithmic implementation. Keeping in view that PFM methods suffer local minima problems and roadmaps has high computational time it was decided to use sample-based techniques in the UTM Simulator and selected PSO for its robust implementation and swift convergence performance. The PSO algorithm has been used in path planning an obstacle avoidance in multi-UAV trajectories data generation steps of this thesis.

Table 2-1 Qualitative comparison of path planning algorithms

Path Planning Algorithm	Pro	Cons
Particle Swarm Optimization (PSO):	• Greater capacity to generate satisfactory outcomes without the need for local searching • Strong tendency towards premature convergence • Not many parameters to adjust • Greater efficacy and probability in locating global optima[122]	• Unable to resolve scattering problems [123] • Memory required for velocity updating [124]
Potential Field Method	• A lightweight, simple method Smooth, optimal paths are generated [112]	• Challenging implementation for real-world applications. • Dynamic environment leads to poor performance

		• Tendency to stick in local minima situations • Difficulty coping with symmetrical obstacles[125]
Probabilistic Road map (PRM)	• Show superior path exploration, which places this option among the successful path-planning approaches for UAVs • Straightforward to understand and implement [126]	• Computational costs are high • Dynamic environments lead to poor performance [125]
Genetic Algorithm	• Accuracy: feasible, near-optimal solutions derive from solutions around large variables and constraints [127] • Applicable with both continuous and discrete parameters • End solutions are not negatively impacted by bad solutions [123]	• Iterate: the number of required iterates increases in line with greater numbers of variables and constraints; generally, far more iterates are needed than for PSO [127]. • May become trapped in local optima [123]

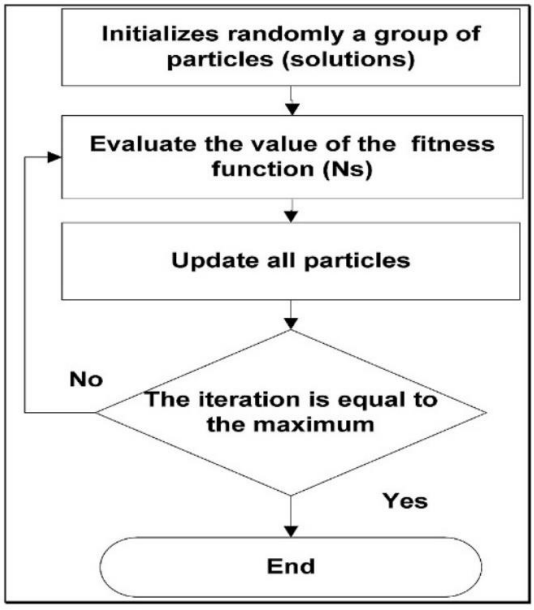

Figure 2-4 Flowchart of a PSO algorithm [128]

2.5 Rule-Based Conflict Management for UTM Scenarios

2.5.1 Review of Conflict Detection and Resolution Methods in UTM

UAVs are revolutionizing many everyday human tasks, including transportation, logistics, and surveying. This is because they open the door to numerous opportunities, such as reducing risks and costs [129] in myriad real-world applications. There will, therefore, be an increasing trend towards UAV deployment [130]. Thus, low-altitude air traffic is expected to rise considerably in the years to come. This fact has been further emphasized by [91], which anticipates the near ubiquity of UAV flight-origin and destination points. To quote Dr. Parimal Kopardekar from the open forum of DASC 2016: “Every home will have a drone and every home will serve as an aerodrome” [131]. One of the prerequisites for the fruitful and safe real-world deployment of UAV fleets, however, is the development of a safe and efficient Unmanned Aircraft System (UAS) Traffic Management (UTM) model [92]. There is a pressing need to implement these services with safety, and thus the path of each UAV must be free from static obstacles, such as elevated terrains and no-fly zones. There is also a need to address dynamic obstacles, such as other UAVs. Conflicts can result in a wide range of negative impacts, both in the air (e.g., delaying or not delivering a service, obstructing airspace for other users, and ultimately collisions) and on the ground, which might involve fatalities. A conflict is defined as the predicted loss of both horizontal and vertical separation between two or more aerial vehicles [63]. It is necessary to develop an efficient conflict-management system that, in turn, must be able to detect conflicting traffic in sufficient time, and perform avoidance maneuvers to resolve conflicts [64].

Separation poses a significant challenge in both the ATM and UTM systems. In ATM, controllers rely on conflict resolution maneuvers to maintain minimum separation between aircraft when they anticipate a conflict. While controllers receive assistance from decision support tools, the task primarily falls on human operators. In UTM, the expectation is that UAVs will ensure their own separations through the use of detect and avoid algorithms, which bear similarities to the onboard Traffic Collision Avoidance System (TCAS) system of manned aircraft.

The evolution and improvement of these collision avoidance algorithms are anticipated to also benefit the ATM system, where aircraft self-separation is currently limited. However In UTM, and also in Air Traffic Management (ATM), Conflict, Detection and Resolution (CDR) methods refer to different layers of "redundancy" (Figure 2-5) as suggested by the International Civil Aviation Organization's (ICAO) Global Air Traffic Management Operational Concept [128]. The first layer is the pre-flight CDR, where off-line conflicts are identified and resolved via flight plans submitted to the UTM, before the actual flight. The second is In-flight CDR, which also considers changing weather conditions, or certain real-time emergencies. Conflict-free paths created by pre-flight CDR might no longer be safe. Therefore, in-flight CDR methods help to moderate UAV movement in real time by providing conflict-free paths.

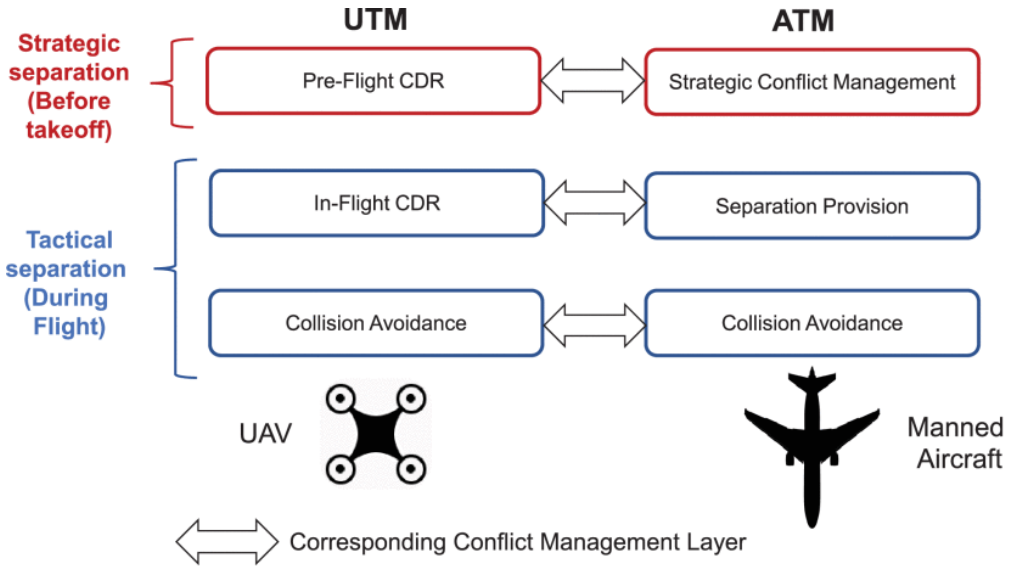

Figure 2-5 Conflict management layers in ATM and UTM [130]

Air Traffic Flow Management (ATFM) for aircrafts has been studied intensively by researchers, to integrate conflict detection and resolution strategies. This problem was addressed by Kuchar & Yang, who surveyed and categorized about 68 such schemes, providing a taxonomy for classifying such methods [132]. A further review paper [133] presented more up-to-date content. The Federal Aviation Administration has, in this context, decreed that Unmanned Aerial

Vehicles (UAVs) must evince capacity for Sense and Avoid; otherwise, permission to operate in civil airspace will be withheld [134].

In order to facilitate smooth integration and maintain appropriate spacing between drones and other aircraft operating within shared airspace, the (ICAO) has established guidelines. These guidelines specify a horizontal separation of 5 nautical miles (NM) and a vertical separation of 1000 feet for manned missions [61]. However, in the case of unmanned aviation, there is no universally defined standard separation distance. Nevertheless, a commonly utilized value in research is 50 meters [62]. The ICAO, moreover, directs that the CD&R capabilities of UAVs must encompass avoidance and detection in dynamic, as well as static environments. Civil UAVs will not be permitted to operate beyond visual line-of-sight (BVLOS) parameters, unless and until such criteria are satisfied [135].

Recent works have proposed several separation methods, such as a 4D (x,y,z & time) trajectory-based pre-flight airspace reservation method [130], modelling of the pre-flight conflict detection and resolution as a Multi-Agent Path Finding problem [136], implicit coordination of drone maneuvers, and use of combined navigation and swirling functions [91]. Nonetheless, these methods are computationally complex and generally appropriate for pre-flight deconfliction. They have limited scalability, and they also require additional (strategic and tactical) deconfliction to provide the required safety level. The work in [137] proposes using a selective velocity obstacle method in tactically de-conflicting UAVs in real time. The study in [138], meanwhile, proposes an adapted optimal reciprocal collision-avoidance approach, which takes care of practical considerations such as navigation inaccuracies, communication overheads, and flight phases, in order to state minimum separation distances for a conflict-free zone. It is thus evident that the above approaches consider the integration of in-flight CDR methods only.

Evans et al.[65] explore the “fairness” implications of decentralized trajectory deconfliction for UTM. The authors used simulation as a tool to explore how a First-Come, First-Served (FCFS) approach to strategic deconfliction in UTM

performed in terms of fairness. The latter is quantified by comparing average ground delay across operators, and by calculating a normalized fairness metric that accounts for operator cost of delay. The idea of a pre-flight rerouting and filling service, meanwhile, was presented by (Balakirshnan *et al.*), where the filling service ensures that new routes do not conflict with other trajectories [128]. The work in [91], meanwhile, presents different conflict-resolution schemes for low-altitude, uncontrolled airspace, and it estimates capacity under various airspace-management scenarios. The authors evaluated various conflict-resolution methods, including ground-delay, hovering, layered assignment, and on-demand descent for single-layer airspace, where all aerial vehicles occupy the same altitude. They also examined multilayers, considering the horizontal layers at altitude levels below or above. The proposed methods were validated and compared, using simulations. A recent UAV conflict-resolution study [133], furthermore, evaluated more than 100 conflict-resolution methods for both manned and unmanned aviation systems. That study comprised an assessment that, in turn, was conducted under a taxonomy derived from surveillance, trajectory propagation, resolution maneuver, type of obstacle, avoidance planning, control, predictability assumption, multi-actor conflict resolution, and optimization, as well as a category based upon methods of resolution. It has, furthermore, been pointed out that a large volume of unmanned aviation still focuses on static obstacles only, and thus needs further development for BVLOS unmanned operations. The authors appreciated that the development of CD&R technologies for unmanned aviation is an urgent matter, given the increasing deployment of drones in urban locales for package delivery, etc. Such drones must, of course, be able to avoid both static and dynamic obstacles.

Aside from these studies, other researchers have focused on sense-and-avoid methodologies. For example, in the Agent Fly project [139], the sense-and-avoid function uses intelligent algorithms and a communication channel provided by onboard wireless data modems, and by sensory data providing information about objects in its surroundings; this involves (e.g.) onboard radar systems or transponders. The work in [140] suggests a reciprocal collision-avoidance algorithm, based on the velocity-obstacle approach. The latter guarantees

collision-free maneuvers, even when the agents are only capable of sensing their environment within a limited field of view. The work in [134] address the need for airspace to be rendered flexible and dynamic in view of airspace demands, equipage and weather conditions. The authors propose three core areas to investigate, specifically: restructuring airspace, adaptable airspace, and generic airspace environments. They also present mid-term and long-term airspace-configuration concepts, which include high-altitude airspace and low-altitude airspace regions for high density. This study may help in devising any future CDR methodologies for UAV dynamic-airspace environments.

The study in [139] addresses the use of priority-based planning to avoid collisions, whereby each robot is assigned a unique priority. The trajectories for individual robots are then planned sequentially, from the highest-priority robot to the lowest, to avoid collision with static obstacles and other robots.

Several ATM schemes have been studied that may lay down a path for de-conflicting UAVs. The work in [141] addresses the use of prior flight-level allocations and ground-holding strategies for ATM scenarios. They suggested using a prior flight-level allocation program to reduce the complexity of air-traffic flow, before applying the ground-holding algorithm to resolve all conflicts, thereby improving the quality of solutions. A study in [142] also deploys shifts in departure times or ground delays to resolve ATM conflicts. Nevertheless, their methodology generates an average departure-time shift of 21.6 to 23.2 minutes, with a maximum 60-minute delay, which would add costs for UAV traffic agencies. This work addressed the problem of rerouting by constructing an alternate trajectory, with the help of M-virtual waypoints along the nominal trajectory in the horizontal plane. These were later reconnected using straight-line segments. The path optimization was undertaken via a simulated annealing algorithm.

The European Network Manager Operations Centre suggests different strategies to deconflict whenever an *en-route* sector is overloaded. These include ground holding, flight-level allocation, and assignment of alternate trajectories [143]. This work has also evinced ground delays of the order of 30 to 60 minutes in resolving conflicts, but it suggests the assignment of a fairness factor to conflicting UAVs.

The objective is to minimize the delays, financial losses and environmental impacts of congestion. A ground-delay approach, meanwhile, is presented in [144], whereby all adjustments are imposed on departure times, in order to avoid conflicts. Furthermore, the authors present an approach to account for uncertainties, which unfortunately results in high delays. The work in [145] explores the implications of decentralized trajectory deconfliction for fairness in UTM. The authors used simulation as a tool to explore how a first-come, first-served approach to strategic deconfliction in UTM would perform in terms of fairness. The latter is quantified, here, by comparing average ground delay across operators, and by calculating a normalized fairness metric to account for operator cost of delay.

2.5.2 Deconfliction Methodology

Few of the preceding studies have suggested hierarchical, phased deconfliction strategies for UAV regimes, using rerouting, ground delay and hovering. To resolve any UAV conflict that appears in the scenarios, this work attempts to present a more realistic picture by associating priorities with various UAV missions, and by applying these strategies phase-wise to achieve fine deconfliction efficiencies. It is observable that, as traffic grows, the pre-flight level allocation is unable to resolve the growing high-density conflicts [130]. Furthermore, this study indicates the need for multi-stage deconfliction by applying the independent strategies one after another, to resolve conflicts.

The proposed rule-based, conflict-management model consists of three main stages (Figure 2-6). The first stage includes strategic deconfliction during the flight plan generation (rerouting). The second stage, pre-tactical deconfliction, applies a ground delay to the agent to resolve the conflict. The third stage corresponds to the tactical deconfliction, whereby the drone hovers or loiters in the last waypoint before the conflict area, until the conflict time window passes. The proposed model combines these methods in a seamless operational flow to enhance deconfliction, in order to cater to more complex and dense traffic operations.

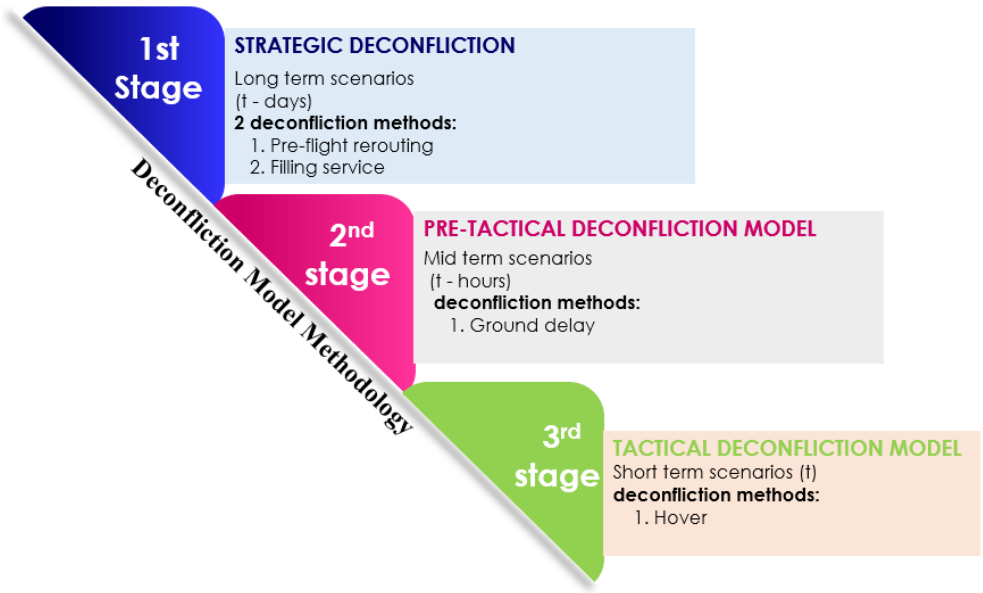

Figure 2-6 Rule-based conflict management for UTM scenarios

2.5.2.1 Simulation Setting and Considerations

The proposed method is assessed and validated through simulation that represents the airspace over Bedfordshire and Buckinghamshire in the UK. It uses nine areas where flights may be restricted, such as airfields, recreational areas, and a prison. Specifically, these areas are:

1. Four airfields, Luton, Cranfield, Halton, and Old Warren.

2. Four recreational areas, including Dunstable, Sandy, Cardington and Graveley

3. Milton Keynes Prison.

The following simulation settings were considered, in order to set the limitations and requirements, and to render our work more realistic, i.e., closer to a real-life scenario:

- Random environmental availability: In order to consider the dynamics in an environment, the availability of recreational areas and dynamic airfields are set randomly during the second and third hour of the simulation scenarios. The areas, and the number of minutes that “disappear” and are inconstant, are selected randomly.

- Random number of flights per hour: A random number of flights is set to fly per hour, in order to force the system to address some conflicts. This number varies between 30 and 40.
- Random departure times: To make the system more realistic, the exact time each agent departs is set randomly.
- Period of simulation runs: The duration of each scenario has been set to one hour.
- Random start and finish locations: The simulation has been set so that the locations of the start and finish points are selected randomly. The simulations address the location map for the whole Bedfordshire area in the UK, and each flight departs and finishes at a random point.
- Random priority levels: Each flight has been set a random priority level of service from level 1 (highest priority) to level 5 (lowest priority). This random prioritization will be used in our deconfliction strategies.
- Constant UAV velocity: The velocity of each UAV agent is considered as constant, and this applies to all UAVs.
- It is important to note that the operational volume concept is still applicable. However, for the purposes of this study, a constant altitude level is considered, specifically focusing on the 2D cruise phase. This choice aligns with the assumptions detailed in section 2.2.2 above. Moreover, the model used in these simulations did not explicitly consider the probability of UAV movement.
- In this study, a three-layered deconfliction management framework is proposed, incorporating a lateral separation distance of 50 meters to ensure safety. This adaptation is based on ICAO's definition mentioned earlier. However, vertical separation has not been taken into consideration, as discussed in the assumptions of the idealized transport model.

2.5.3 Simulation Scenarios

Three simulation scenarios were run in MATLAB for the Bedfordshire environment, as explained earlier. The number of UAVs flying per hour in these simulations ranges between 30-40. Trajectory data comprise a set of waypoints,

and conflict is defined when two or more agents are present at the same coordinates at the same time.

2.5.3.1 First Hour of Simulations – Static No-Fly Zones

The first hour ran between 8 am and 9 am. During this period, UAVs were not allowed to fly over the nine defined obstacles that remained static, and which did not open their respective airspaces. A total of 32 flights were scheduled during this hour of operations, and all of these flights departed and landed at random locations. The PSO-based algorithm found a feasible path solution for all the required trajectories. During this period, 14 conflicts were found. The conflict heat map is presented in Figure 2-7. In this figure, the heat map of the region is created based on the conflict level definition, which is determined by counting the number of UAVs passing through specific coordinates (x, y) within a time horizon of 5 minutes. The heatmap presents clusters or groups of data points with high UAV counts, indicating high-risk conflict zones, which are highlighted in red. Conversely, areas with lower UAV counts are depicted as the lowest conflicted zones, highlighted in green.

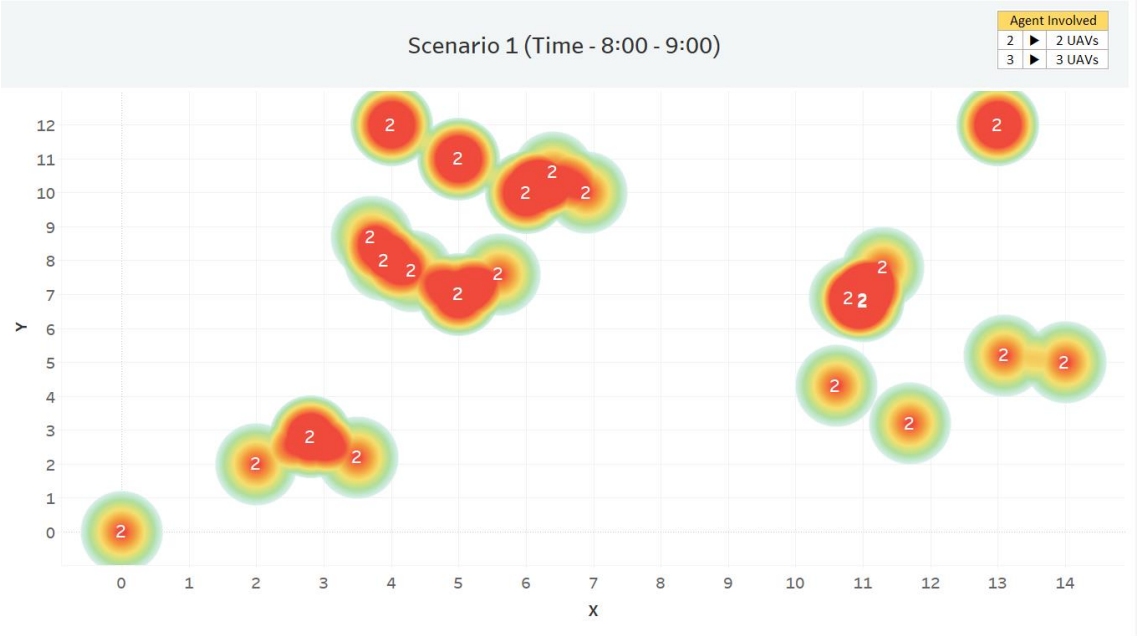

Figure 2-7 First hour simulation - conflict heat map

2.5.3.2 Second Hour of Simulations – Dynamic Recreational Areas

The second hour of simulation ran between 9 am and 10 am. The difference with the previous simulation was that the no-fly zones corresponding with the recreational areas were dynamic. This dynamism was achieved by making the availability and unavailability of these areas random, during the whole hour. Therefore, some agents could fly over these areas at some point, during this hour. The recreational areas of Cardington and Graveley also opened their respective airspaces. A total of 39 agents were simulated during this hour. During this period, 26 conflicts were found. The conflict heat map is presented in Figure 2-8.

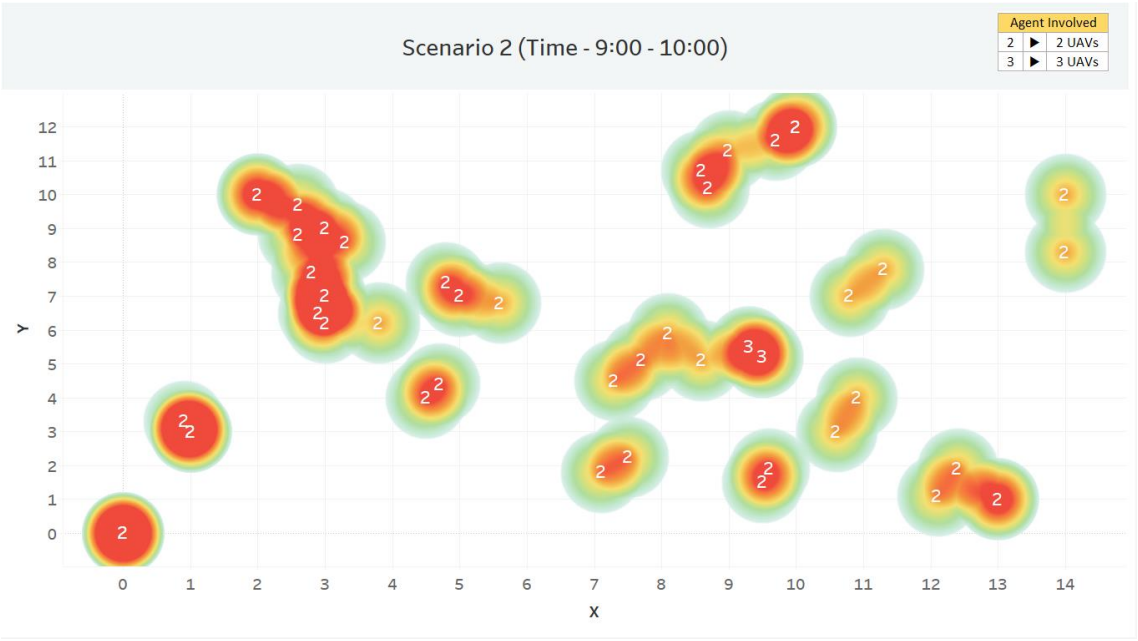

Figure 2-8 Second hour simulation - conflict heat map

2.5.3.3 Third Hour of Simulations – Dynamic Recreational Areas

The third hour of simulation ran between 10 am and 11 am. The heat map is shown in Figure 2-9. The difference with the previous simulation is that the no-fly zones corresponding with airfields were dynamic. This dynamism was achieved by making the availability and unavailability of these areas random during the whole hour. Therefore, some agents could fly over these areas at some point during the hour. Moreover, the Luton and Old Warren airspaces were opened for traffic. A total of 35 flights were simulated during this hour, and 18 conflicts were found.

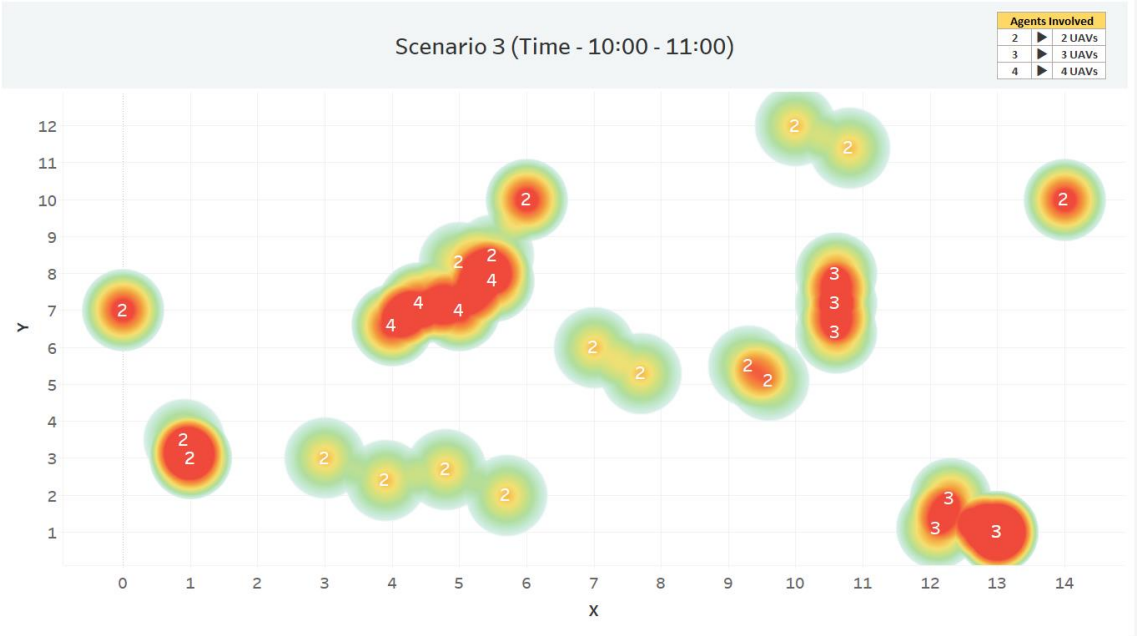

Figure 2-9 Third hour simulation - conflict heat map

A further key figure of merit has been created in this work, which corresponds to the conflicts per hour of each simulation, and the number of agents in the sky at that time. Using three hours of simulation data, the relevant statistics are presented below (Figure 2-10). In the vertical axis, we observe the number of conflicts, whereas in the horizontal axis one may find the time scale. The size of each marker, and the numbers enclosing the markers, denote the number of agents (1-20) flying at that specific moment.

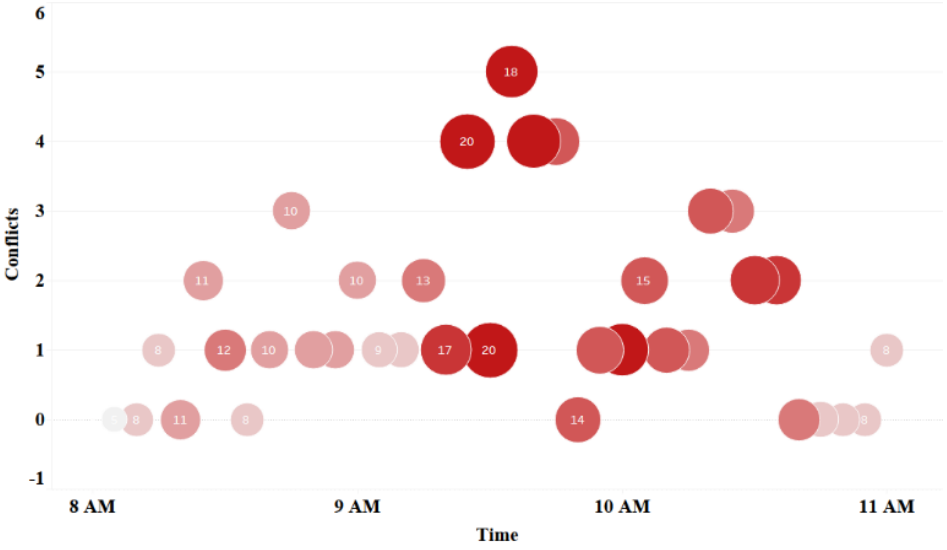

Figure 2-10 Number of conflicts and agents in the sky from 8AM to 11AM

A trend analysis has also been conducted, based on the number of flights and conflicts for three hours of the simulation, as shown in Figure 2-11.

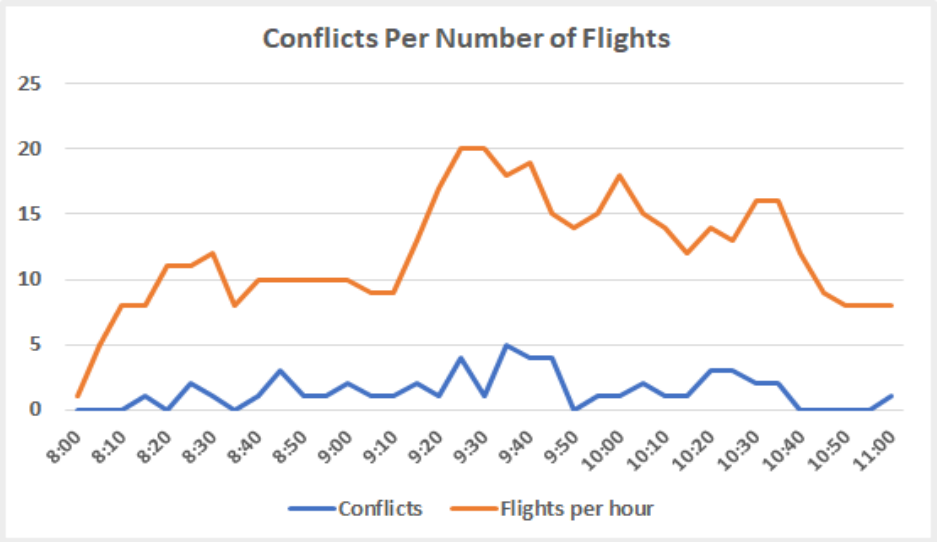

Figure 2-11 Conflicts per number of flights

It can be inferred from both (Figure 2-10 and Figure 2-11) that the greater the number of agents in the sky, the greater the chances of a conflict, and there is also the possibility of more than one conflict at a particular moment. (In fact, we observed a maximum of five simultaneous conflicts.) It is also observable that, as flights are added in the airspace environment for all three hours of simulation, the number of conflicts grows in a polynomial pattern.

An important correlation exists with the change in the environment in the second and third hours of simulation, in which the airspace environment was opened for some recreational areas and airfields. There was a more modest rate of increase in conflicts, in comparison with the first hour of simulation, when the airspace remained closed for all dynamic recreational areas and airfields. This seems a more credible picture *vis-à-vis* the real world. The rate of conflict tends to decrease with the availability of more areas to fly in, whenever certain dynamic areas are opened.

From Figure 2-11, it can be seen that between 9:20 and 10:40 hours, during the simulation, there was a steady increase in the number of flights, resulting in some regional conflict peaks. This increase was due to presence of more UAVs in the

air from previous hours, plus new incoming flights, and also their interacting trajectories.

From the first-hour simulation trajectory and heat maps, it can be seen that the corridor between two zones, especially (3,7) and (11,7), provides the airspace for most flights to pass through. This increases the likelihood of conflict and results in “red-hot conflicted zones,” where more than two UAVs conflict simultaneously. The second hour of simulation allowed the airspace above the recreational areas of Cardington and Graveley to open for flights. This resulted in more space to fly over, and in turn, this caused an even spread of conflicts in this comparatively wider corridor, in contrast to the first-hour simulations. This phenomenon is visually apparent, as there are fewer red-hot areas on the heatmaps. The same phenomenon was observed in the third-hour simulation data, when Luton and Old Warden opened their airspaces. Nonetheless, some new, orange heatmap areas, or less-conflicted zones, do appear, due to the addition of more incoming flights, and pre-existing flights, in the airspace during these later simulation hours.

2.5.4 Deconfliction Methodology

The next section discusses the deconfliction models proposed, and applied, to the conflicts in this dense airspace environment, the goal being strategically to deconflict any number of trajectories within the same environment. An important consideration in dealing with conflicts with multiple agents, involved simultaneously, is that deconfliction should be applied to the agent with the lowest priority level. The proposed deconfliction model comprises three different stages. We propose to apply each stage before departure, and during UAV flight.

2.5.4.1 Deconfliction Model: First Stage

The first stage Figure 2-12 is ideally applied days before departure. This stage comprises two different methods, namely, pre-flight rerouting and the filling service. Pre-flight rerouting involves changing the route when there are two trajectories at the same point at the same time, thus avoiding this conflict zone. The filling service checks that this new route does not conflict with other

trajectories. The flow chart for this stage is presented below, and it will be connected to the rest of the chain in subsequent sections.

This approach either modifies the Extended Flight Plan (EFPL) for all the conflicts in a multilayer (based on two designs of the airspace: the single-layer airspace, where all UAVs occupy the same altitude, and multilayers whereby UAVs have different altitude capabilities [91]), or alternatively, it causes the cancellation of an existing EFPL, if conflicts still exist. An extended flight plan incorporates additional information beyond trajectory points, including speed, aircraft mass, and the predicted climb and descent profiles for a specific flight.

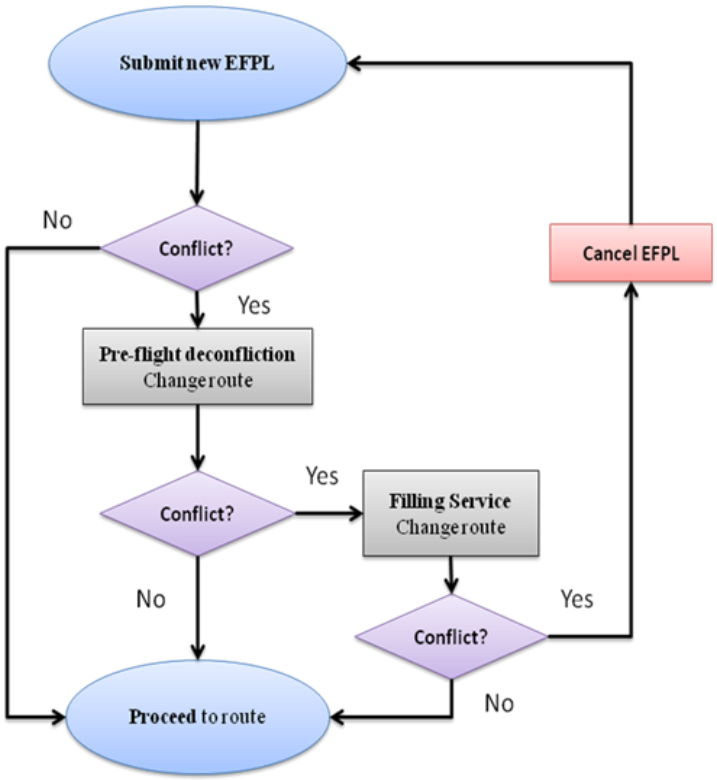

Figure 2-12 First stage of strategic deconfliction

2.5.4.2 Deconfliction Model: Second Stage

The second stage of deconfliction as shown in Figure 2-13 is achieved by applying ground delays to lower-priority UAVs before departure. The maximum delay could extend from two to three minutes on the ground. The proposed

process flow is divided into two branches: one for a single layer of airspace, and the other applied to multiple layers of airspace. The ground delays are introduced in the case of conflicts in single or multilayers, with the order to maximize delay time until all the conflicts are resolved. Otherwise, the model leads to the cancellation of an EFPL. If no conflict exists, the agents are allowed to proceed as per their routes.

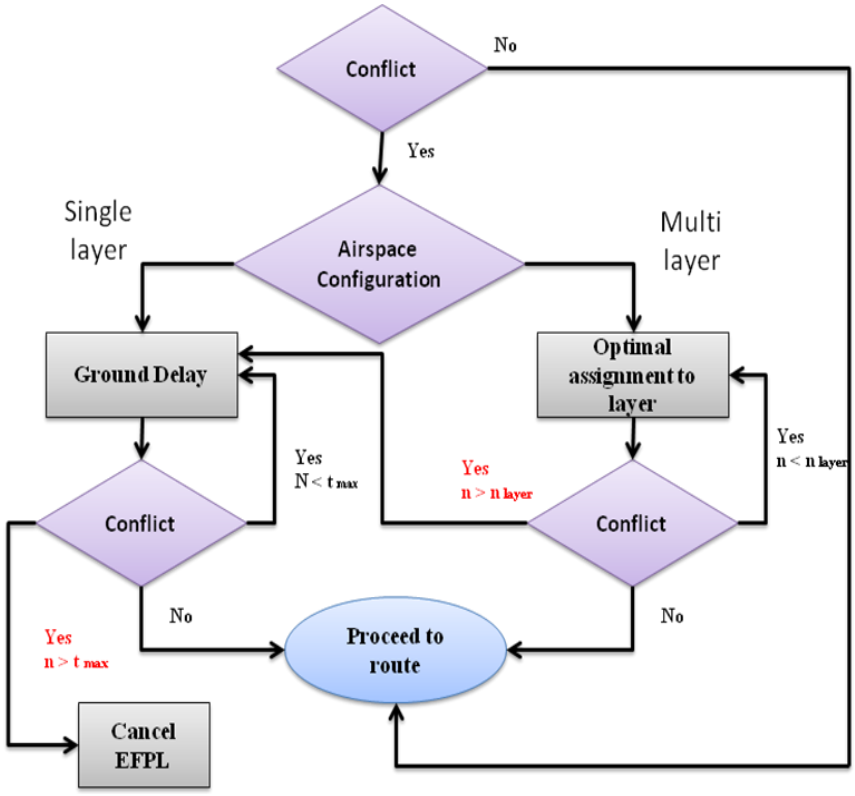

Figure 2-13 Second stage of strategic deconfliction

2.5.4.3 Deconfliction Model Third Stage

Hovering is the third stage of deconfliction, applied around the departure time itself, as depicted in Figure 2-14. The process of hovering starts at a waypoint preceding the conflicted zone. This introduces a pause in the process until the conflict is resolved. This stage is highly dependent on one factor, namely, the battery endurance of the UAV. Battery time is the biggest constraint when applied to any UAV, and this factor needs careful consideration before adoption. Secondly, hovering at a previous waypoint is a solution that only rotary-wing agents may adopt, rendering those UAVs more maneuverable and easier to

deconflict. We recommend that fixed-wing UAVs be allocated higher priority levels, however.

Hovering is applied in the case of conflicts in single or multilayers. In the event of conflict, the UAVs in the air are subject to hovering at the last waypoint, until the expiration of safe battery time (tbat), or landing, if that time expires. If the conflict is resolved, the other agents in this conflict are allowed to proceed to their designated routes.

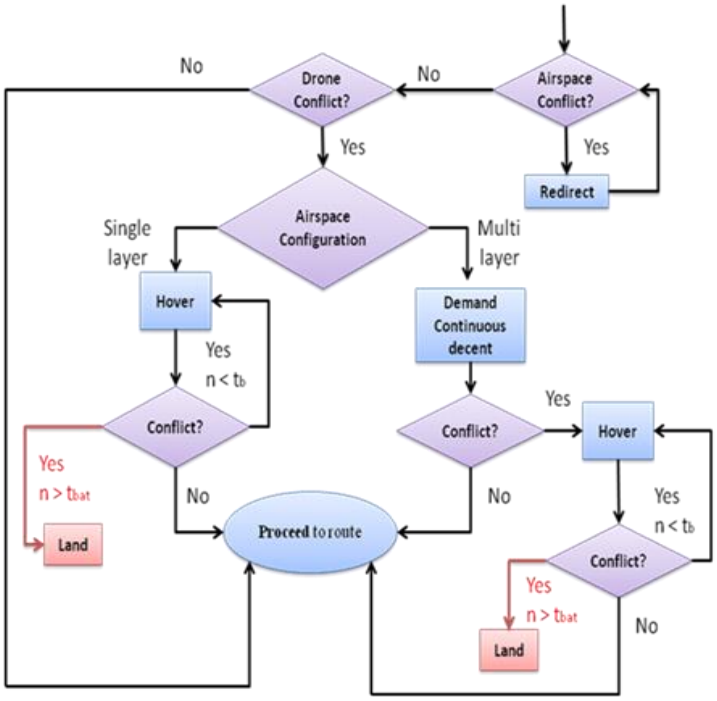

Figure 2-14 Third stage of strategic deconfliction

2.5.5 Application of Deconfliction

In order to demonstrate proof of concept, the methodologies described in the previous section were applied to 58 conflicts that appeared during the simulations. The next section will present the results of the reroute, ground-delay and hover strategies, with the help of examples.

2.5.5.1 Rerouting

Rerouting is the primary method of deconfliction proposed here: it is based on modifying the route of the conflicted UAV, days before the flight mission. We also recommend applying this strategy to UAVs with less critical tasks or lower

priorities. In the example below, a conflict between two independent airspace users was resolved through rerouting. These trajectories were uploaded as EFPLs by two different airspace users. The first user wished to fly from point (0.0) to point (10,12) (blue line), while the second user wished to fly from point (0,7) to point (5,1) (orange line), as shown in in Figure 2-15.

The orange user departs 30 minutes after the blue user. It can be seen that the two trajectories cross at point (x, y) = (2.62, 3.84), at the same moment (43.1). Today, UAV traffic numbers are increasing, and complex UTM operations are becoming a reality. Fairness and equity are essential in the UTM framework of the ICAO, which states that “access to the airspace should remain equitable” [146]. To deconflict, first, both users’ priority levels must be checked. The blue user has a lower priority level, and therefore, this user must be rerouted. To reroute a trajectory, a virtual obstacle is created for the trajectory that requires deconfliction, at the point of conflict. The conflict area, which is represented by a red circle in Figure 2-16, demonstrates the avoidance mechanism of the algorithm, with a new, deconflicted green route.

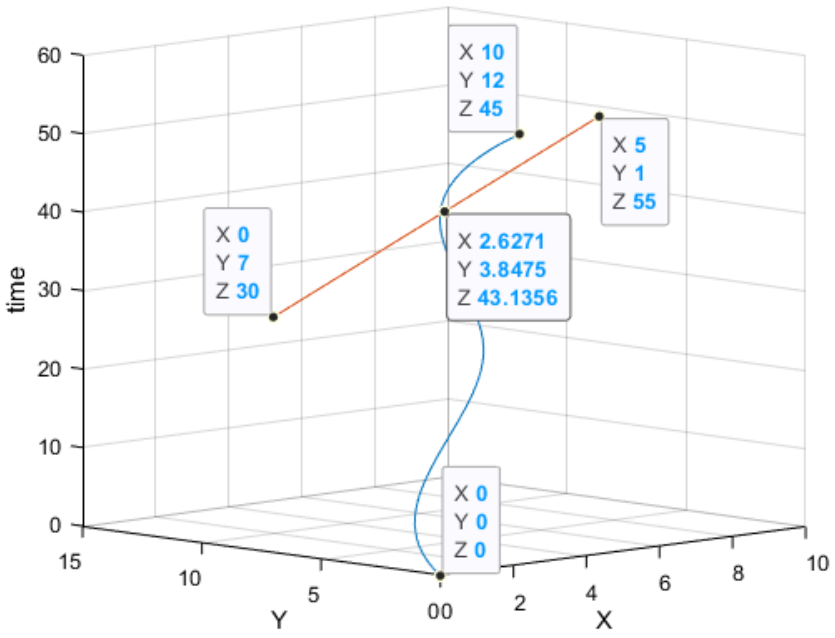

Figure 2-15 Rerouting Example: Confliction case

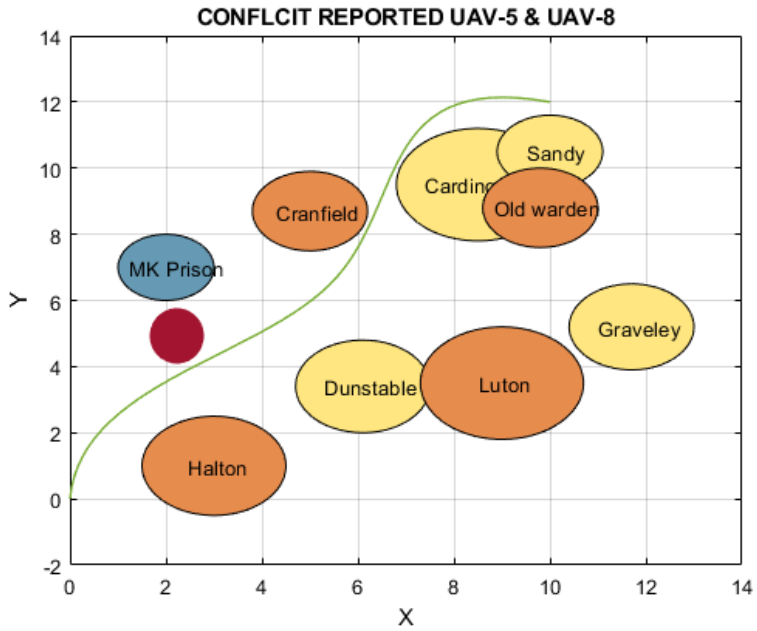

Figure 2-16 Rerouting Example: Deconfliction case

Moreover, an assessment is performed to establish whether this new trajectory conflicts with another agent. This process of the filling service works in the same manner as the first rerouting service, explained above. The deconfliction method applied in this scenario resolves the problem, as both UAVs can cross the point of conflict at different timings, thus avoiding conflict.

2.5.5.2 Ground Delay

The second layer of the deconfliction method is generally applied a few hours before departure. Assuming that there are two UAV flight plans, the first trajectory is planned three days before departure, and the second trajectory comprises an emergency operation, shared twenty minutes before departure. Both flights have a conflict at the same time, and at the same point. In this case, rerouting was not planned. Therefore, a ground delay of three minutes is applied to the lowest-priority flight This resolves the conflict and ensures that a new conflict does not appear, due to the same ground delay.

2.5.5.3 Hovering

The third layer of the deconfliction method proposed in this work is hovering. When none of the above deconfliction strategies are suitable for a specific

situation, hovering is the final option. This technique is limited to rotary-wing agents, and it is dependent upon critical factor considerations, such as battery life. Battery endurance, in turn, depends on maximum take-off weight, battery capacity, planned trajectory/flight time, and so forth. Nevertheless, we did not consider this in the simulations for the present study, due to the simplicity of the simulated scenarios. Hovering maintains a UAV in static mode for two minutes, effectively paused until the conflicted trajectory passes.

Overall, with the proposed, layered deconfliction methodology, the solution proved able to deconflict every flight in a strategic phase for all 58 conflicts, as shown in Figure 2-17.

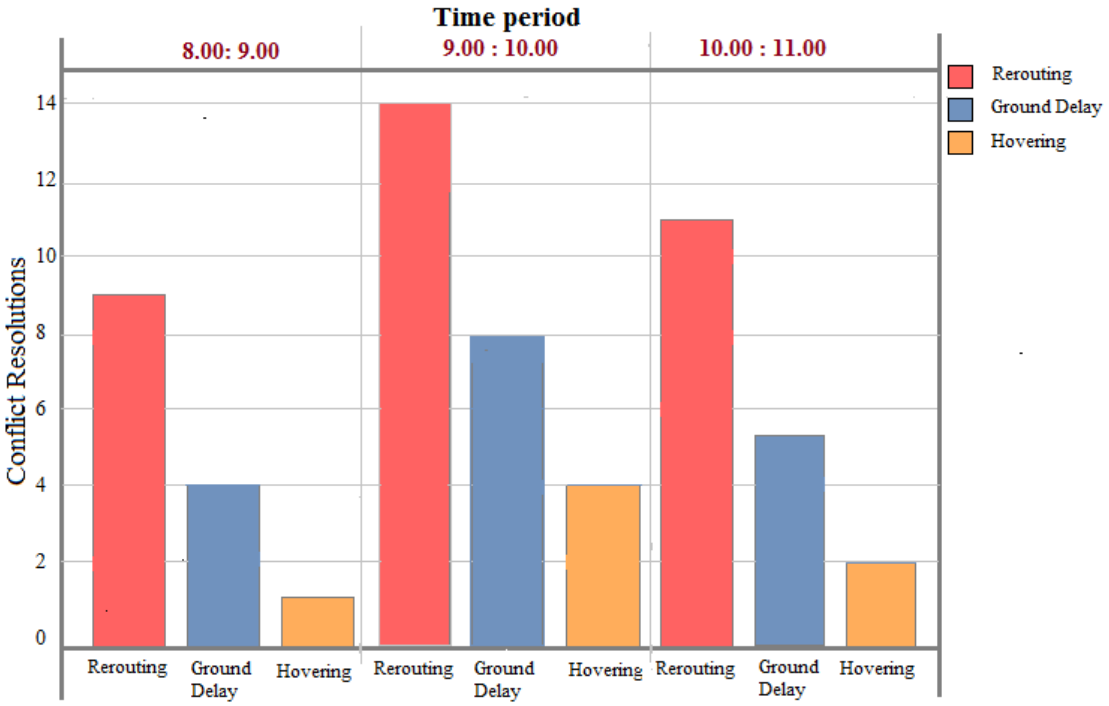

Figure 2-17 Conflict resolution by process

In the first hour (8:00-9:00) of the simulation, with no-fly zones, a total of 32 agents were simulated. Meanwhile, 14 conflicts were resolved, followed by 26 in the second hour (9:00-10:00). A total of 39 flights were scheduled during this hour of operations, which encompassed the dynamic availability of recreational areas. All 18 conflicts were addressed in the third hour (10:00-11:00) of the simulation, and there were 35 simulated flights. Since the airspace was heavily loaded (with a

greater number of flights) between 9:20 and 10:20, we observed a greater number of conflicts during this period. The rerouting mechanism was the most successful method for deconflicting the airspace in these scenarios, followed by ground delay and hovering.

As noted earlier, the airspace was narrow during the first hour of simulation, due to airspace restrictions, and this provided less room for rerouting. Nevertheless, the second and third hours of simulation provided more airspace volume in which to apply the rerouting mechanism, as these periods saw some airspaces opened dynamically for flights.

It was observed that ground delay was the second most successful method, as hovering (the third) came with additional constraints of battery capacity. Moreover, the presence of a static UAV in the air would add to the complexity of the airspace by obstructing other flight paths: this makes deconfliction difficult, as compared to ground delays. More studies, with different traffic volumes and environments, however, would be highly illustrative in examining how traffic volume might affect strategic deconfliction.

2.5.6 Deconfliction Methodology Results Discussion

The proposed deconfliction methodology has evinced high-end gains by resolving all the conflicts in a layered fashion, i.e., by applying independent strategies one after the other. It was observed that for the first hour of simulation data, rerouting was able to resolve 64% of all conflicts reported. Ground delays were able to resolve only 29% of total conflicts. The hovering mechanism, meanwhile, only helped resolve the remaining 7% of conflicts.

The second hour of simulation data, pertaining to dynamic recreational areas, witnessed 54%, 31% and 15% resolution, by rerouting, ground delays and hovering, respectively. These figures were closely mirrored by those for the third-hour simulation data, which evinced 61%, 28% and 11% resolution for the three strategies, respectively.

Airspace standards determined the prioritization of operations [146]; i.e., UAV flights with emergency status had priority over other flights. To reroute the UAV

with the lowest priority within a conflict, a virtual obstacle was created with a 0.5km radius at the point of conflict. It was noted that the proposed rerouting methodology was able to resolve almost 59% of the conflicts arising during the various static and dynamic environmental scenarios. Our work also achieved a reasonable grounding delay, ranging from three to five minutes, in resolving conflicts arising from emergency-mission trajectories generated some minutes before departure. This accounted for about 29% of all conflicts.

Hovering, in the order of two minutes waiting time, was applied at a tertiary stage in the proposed methodology, in the same horizontal space, at a waypoint just before the point of conflict: this resolved about 12% of remaining conflicts. The drawback of this approach was the additional power requirement, which could even lead to an emergency landing while waiting at airfields. In order to resolve this problem, we recommend allocating high priorities to low-energy drones, while preventing hover-like maneuvers for these vehicles.

Few previous studies have provided an efficiency assessment for a layered, multi-stage deconfliction methodology, as applied to a UAV regime. Nevertheless, we may analyze our statistics in comparison with those of [141], which applied pre-flight level allocation or rerouting to resolve conflicts for one day of European airspace data, with residual conflicts solved by a ground-holding phase. The pre-flight level allocations or rerouting resolved 65% of conflicts, and the remaining 35% were managed via the ground-holding phase. The results provide similar metrics, whereby rerouting resolved about 59% of conflicts, alongside ground holding (29%) and hovering (12%). The authors' work also evinces a maximum ground delay of three to five minutes, for almost 90% of their flights.

The potential downside of this study's deconfliction strategy would be loss of flight efficiency. Flights would be delayed, either due to a longer alternate flight path, or because of delays on the ground, or in the air due to hovering. The authors of [142] suggest using an extension in travel time as one of the performance metrics in evaluating the efficiencies of deconfliction strategies. This metric also directly affects the operating costs of any travel plan. Since the most prominent rerouting

strategy is based on modifying the path of single, low-priority UAV, it is assumed that this will not lead to longer extensions in travel times for the other UAVs in the air.

2.5.7 Conclusion and Future Recommendations

It has been observed, via simulations, that as the number of agents in the sky increases within a particular time slot, more conflicts are reported. The present writers observed a maximum of five simultaneous conflicts, when a maximum of 18 agents were in the sky. With regard to the trend analysis, it was observed that this increase followed a polynomial trajectory. The effects of introducing static and dynamic environments indicated that the rate of conflict tended to decrease when dynamic areas were opened up, as a result of more available flight space being provided. The efficiency of the proposed, three-layered deconfliction model, based on rerouting, ground delays and hovering, was evaluated by applying these approaches to the conflicts addressed above. It was observed that rerouting was the most fruitful strategic deconfliction method, accounting for 58.6% of resolutions, followed by ground delay (29.3%). Hovering, meanwhile, merely accounted for 12.1% of resolutions. It was also seen that ground delays, in the order of three to five minutes, were sufficient to resolve some conflicts. A hovering time of two minutes resolved the remaining conflicts, although at lower percentage rates.

Finally, one may note how this project might be enhanced in the future. Several random inputs were applied to the system, with the exception of the velocity of each agent. In future, the speed of UAVs might be changed randomly, to study the effect of changing speeds on the conflicted zones. Although the agent or UAV localization model created and used in this project is two-dimensional (2D; X,Y), a further avenue of future research might involve adding one more dimension (i.e., Z for altitude). This would illustrate how one more degree of freedom would affect system performance. The effect of various environmental scenarios can be studied, in addition, by conducting more simulations, increasing the number of UAVs, and introducing more dynamic areas.

3 MODELING AND CHARACTERIZATION OF TRAFFIC FLOW PATTERNS AND IDENTIFICATION OF AIRSPACE DENSITY FOR UTM APPLICATION

3.1 Introduction

In recent times, the civilian applications of UAVs have risen across diverse fields due to their versatile functionality, automation capabilities and low cost. The application areas of UAVs for both consumer and industry include business and marketing, surveying and monitoring, emergency services and healthcare, delivery services, gaming and sports telecasting, news reporting, the education sector, military, security, and many other fields [147], [148]. Although UAVs have many advantages, they face limitations if their routes are not managed and operated by a high-level automation system. In the absence of such a system, flight operations in airspace can imperil the security, safety, and privacy of the public. To carry out UAV missions and other manned aircraft operations safely, a systematic integration of UAVs is required in the National Airspace System (NAS) before using them for civilian purposes [14].

Factoring in the above concerns, unmanned traffic management (UTM) is designed to carry out the management of UAS operations economically, efficiently and safely, and it is considered a subset of air traffic management. It is also defined as a system of systems (SoS) developed using users' cooperation as well as their systems [38]. The ultimate objective of UTM is to carry out efficient and safe low-altitude aerial operations by providing different services. These UTM services include spacing and sequencing, dynamic geo-fencing, route planning and re-routing, dynamic configuration and airspace designing, contingency management, terrain avoidance, severe wind and weather avoidance, separation management, and contingency and congestion management [149]. Since a number of the above services rely on trajectory data, the field of trajectory data mining and data analytics is seeing a growing interest in the UTM context.

The aviation industry uses flight-tracking data heavily and conducts research using advanced data analytics tools and technologies with an aim to characterize the air

traffic behavior for monitoring airspace, managing air traffic flow, carrying out performance assessment and improving airspace design [150]. Oliver et al.[151] detected and identified particular events in a huge amount of historical aircraft-trajectory data, and it falls under the scope of information extraction and knowledge discovery. Olive and Basora [152] highlighted the usage of data mining in the ATM context by suggesting that it helps sort out the desired information from the huge amounts of data generated daily by various stakeholders of the ATM systems. This information sorted from the generated data is beneficial for different stakeholders for specific purposes: an air navigation service provider (ANSP) may enhance capacity indicators; commercial stakeholders like airports and airlines may improve short-term predictions and optimize their operations; academic stakeholders may design better safety models and make better risk estimations; and air traffic control (ATC) training centers may carry out more realistic simulations.

Considering that ATM and UTM fundamentally differ in their approach to airspace modeling, ATM focuses on managing manned aircraft in a structured and primarily centralized manner, often over vast expanses of airspace and altitude ranges. UTM, on the other hand, is tailored to efficiently oversee a large number of drones operating primarily at low altitudes, emphasizing flexibility, dynamic allocations, and decentralization. Drawing inspiration from previously published works on the classification, analysis, and characterization of traffic flows in the ATM domain [153]–[158], this thesis employing an unsupervised machine learning algorithm, specifically DBSCAN, to discern spatial and temporal UAV trajectory patterns. The characterization of UTM traffic flow is further refined using three innovative quantitative metrics: Distance from the Centroid, Distance to Complete the Mission, and Time to Complete the Mission. These metrics play a pivotal role in assessing deviations in UAV missions and their subsequent impact on the safety and capacity of the UTM airspace. Additionally, this study presents an estimation of airspace density by comprehensively analyzing the entire airspace. Given that traffic density is a traditional measure of air traffic complexity, this analysis seeks to pinpoint regions of interference that may be viewed as potential congestion points, thereby raising safety concerns.

Building upon the groundwork laid in Chapter 2, this chapter leverages the aforementioned simulation framework to design realistic UTM multi-services mission scenarios. These scenarios encompass not only the weather conditions previously modeled but also incorporate static and dynamic obstacles, thereby creating a more challenging and lifelike environment for simulations. Furthermore, the inclusion of random trajectories of recreational UAV users enhances the complexity and realism of the scenarios. The resulting dataset of densely populated trajectories is then utilized to characterize UTM traffic flows and develop congestion contours using a kernel density function. By incorporating these additions, the objective of this chapter is to observe traffic flow behaviors under a broader range of conditions, facilitating a deeper comprehension of UTM system performance and identification of potential areas for improvement. The contributions of this chapter are outlined as follows:

1) Realistic UTM multi-services mission scenarios are designed, incorporating the effects of weather conditions and static and dynamic obstacles along with random UAV trajectories of flying hobbyists (recreational users) trajectories.
2) A UTM flight-trajectory data analytic framework is developed for the identification of UAV trajectory patterns at the spatial and temporal scale that uses an unsupervised machine learning (DBSCAN) algorithm.
3) Key Characterization Parameters (KCP) for modeling UTM mission-data traffic flows are introduced. These metrics are evaluated for worst-case weather scenarios and dynamic airspace constraints, in order to present deviations in UAV missions.
4) A Gaussian kernel density estimator–based airspace congestion analysis is performed to identify hot zones and regions of interference.

The rest of the chapter is organized as follows: Section 3.2 presents related work and some background on the techniques used in this study. Section 3.3 discusses the proposed methodology of this work. Section 3.4 presents the simulation scenarios and results. Section 3.5 discusses the airspace traffic pattern and density. Lastly, Section 3.6 provides a conclusion of this chapter and some guidelines for future work.

3.2 Related Work

3.2.1 Overview of Trajectory Data Mining Techniques

Data mining helps extract beneficial information from enormous datasets and is commonly known as knowledge discovery [159]. Likewise, the primary objective of trajectory data mining is to discover relevant knowledge from trajectory datasets on traffic abnormality, travel behavior and movement patterns. Furthermore, trajectory data mining has two main goals: description and prediction. Description focuses on finding human-interpretable structures while illustrating the data, whereas in prediction, some variables are used in the data to ascertain the future values, unknowns and other required variables [160]. Trajectory data mining has been regarded as a vibrant research area across various domains. Zheng [161] used data mining techniques to classify trajectory data into four categories: mobility of natural phenomena, mobility of transportation vehicles, mobility of animals, and mobility of people.

Several other studies have been conducted in the recent past that are based on air traffic datasets extracting valuable behavioral information of moving objects like animals, vehicles, and people [162]. Recently, many airlines established different divisions for analyzing flight safety and related issues based on historical data analysis. Additionally, many airlines are exploring new methods based on data science for improving their airline safety by analyzing traffic events from historical data [163]. Due to the availability of huge amounts of trajectory data from gadgets and other applications that use GPS, a number of surveys related to trajectory data mining have been conducted [164]. Parent et al. [165] provides an analysis of mobility data management, listing by discussing the main techniques for mining, enriching, building and extracting knowledge from trajectory data. The survey by Kong et al. [166] presents trajectory applications and data from travel patterns, travel behavior, trajectory data service description in terms of transport management, and other aspects. On the other hand, the survey by Bian et al. [167] proposes a set of trajectory clustering techniques, classifying them into three categories: supervised, semi-supervised and unsupervised.

Clustering is an unsupervised learning process. It is considered a first-tier trajectory mining method to determine heterogeneity and homogeneity based on data properties, and thus similarities within a trajectory dataset are revealed by dividing the trajectories into various clusters [168]. In a nutshell, the features of trajectories' movement must be same within a cluster, whereas features vary between clusters. Generally, each trajectory is represented by a feature vector, and later on, the similarity between trajectories is measured by finding the distance between their feature vectors [161].

To design a trajectory-specific clustering model, many efforts incorporating the features of trajectories have been made, and most of them are probabilistic and statistical models. Smyth and Gaffney [169] proposed a clustering approach based on mixture models that combines trajectories that are produced by a common representative trajectory with the addition of Gaussian noise. Similarly, Alon et al. [170] used a Hidden Markov Model (HMM) to model clusters because it fits the trajectories best, and trajectories are designed as sequences of transitions between different points. Clustering algorithms based on the similarity functions determine similar direction, similar source, similar route, and similar start and destination. In this regard, Maimon and Rokach [171] offered a detailed discussion on how to apply similarity functions and distance that further help in determining cluster membership.

Two well-known algorithms have been mentioned in this work. The first one is OPTICS (ordering points to identify the clustering structure) [172], and the second is DBSCAN (density-based spatial clustering of applications with noise) [173]. T-OPTICS [174], an extension of OPTICS, was designed by describing a spatiotemporal distance for clustering and comparing trajectories. On the other hand, ST-DBSCAN [175] was developed as an extension of DBSCAN to improve the identification of noise and clustering using two additional parameters. Depending on the similarity function and the goal of analysis, trajectory clustering can be applied on sections of trajectories or on whole trajectories. The route of trajectories is not important, so they can be clustered if their similarity function is validated on the basis of their similar origin and similar destination [176], [177].

Classification is the second trajectory data mining method. As it is a partially supervised or supervised learning process, it is different from clustering [178]. The classes of classification must be predefined. Moreover, a training set of objects is required to be pre-labelled based on their class. Two steps are needed for a typical trajectory classification algorithm. In the first step, a set of distinctive features are extracted for training an already existing standard classification model such as decision trees, support vector machine, nearest neighbors and logistic regression. In addition, the properties of the trajectory that suit best to describe different classes of trajectories are determined. In the second step, a significant standard classification model is selected, which is then applied to the already extracted distinctive features.

Many comparative studies have been done on various standard classification models [179]. These classical algorithms are directly applied for trajectory classification in most cases. Bolbol et al. [180] used Support Vector Machine (SVM) for transportation mode classification. The authors used statistical methods and assessed the distinctive power of some features of common transportation means such as, train, bicycle, subway, walking, and private car. They also determined the acceleration and speed of these means of transportation. In most cases, some pre-processing such as clustering, segmentation and statistical analysis were carried out to classify trajectories that produce those features required for classification.

Zheng et al. [181] proposed a change point–based segmentation method to divide each trajectory into distinct segments of non-identical transportation modes. The work identified a set of features that are not affected by altering traffic conditions. Patel et al. [182] emphasized on improving prediction accuracy by adding time duration information that helped differentiate dynamic objects moving at different velocities. Lee et al. [183] extracted sub-trajectory and regional trajectory features using SVM-based classification algorithm.

There are also other works related to the discovery of movement patterns hidden in trajectories. Mazimpaka and Timpf [184] discovered the spatial and temporal aspects of these mobility patterns. Nasreen et al.[185] also associated the nomenclature for these movement patterns that include periodic patterns,

associations, sequential rules, subgraphs and frequent items. Wachowicz et al.[186] classified the movement patterns into three different kinds: swarm, convoy and flock. According to this work, a group of objects is referred to as a flock if they move together for at least t successive timestamps and their locations can be observed in a disk of radius. A flock and convoy are quite similar to each other, with a convoy being dissimilar only for the disk shape. As long as the positions of the moving objects are clustered, any disk shape can be formed by a convoy pattern on each time slice. These patterns are clustered usually by density-based clustering with minimum object number and maximum neighborhood distance [187]. The requirements for a convoy are further relaxed by a swarm, and in this case, no object positions are required to be clustered in each time slice [188].

In every case involving swarms, convoys and flocks [160], collective pattern mining is done by density-based clustering, which is considered the most common method. The next section discusses the literature survey conducted in the field of data analytics for air traffic flow identifications.

3.2.2 Flight Data Analysis and Traffic pattern Identification

The availability of historical traffic data has led to attention from research community to identify the trajectories and properties of air traffic [189]. For getting insights about the behavior of dynamic targets like animals, people and vehicles, sensing technologies have already been used in the past [190]. The usage of flight-tracking data in the aviation industry for applying advanced analytics methods has been investigated in several studies. In [150], the air traffic flow was characterized for better traffic flow management. The same type of study is conducted in [191], [192] for better airspace design, airspace monitoring and performance assessment.

In [153], potential events were detected and identified by assessing the deviations in recorded trajectories' data. This study determined the efficiency of aircraft operations in a complex systemized terminal maneuvering region. The performance of aircraft operations is assessed in the area of London's multi-airports due to actual air traffic flow and standard route structures. Basically, this work falls under the domain of traffic pattern recognition and flight data analysis,

particularly in the terminal maneuvering areas (TMA). The methodology followed in this work is comprised of two modules. The first module is termed as data-driven 4D (x, y, z, t) adherence calculation process. The ultimate objective of this process is to assess trajectory deviation, identify recurrent patterns and thus assess the 2D interdependencies of the standard routes and finally determine the high-demand routes. In the second module, concurrence events that may result in conflicts and can cause a lack of adherence are determined. Nonetheless, the above study has been able to give some insights about air-traffic flows by identifying their patterns, but more emphasis is on TMAs such as airports. Furthermore, it does not cover any UAV traffic flow within a densely populated low-dynamic airspace.

Murca et al. [154] applied various clustering methods to learn traffic flow patterns in both spatial and temporal dimensions. The authors analyzed these patterns for complex and super-dense metroplex airspace and estimated the predictability, capacity and efficiency of the metroplex. They also characterized the dynamics of air-traffic operations. Bombelli et al. [193] used clustering to identify general routing structures in US airspace enroute trajectories. Amerson et al. [194] applied trajectory clustering methods to estimate the effects of convective weather on flow rates and dominant routes between New York and Fort Worth centers. The ultimate goal of the above studies was to develop high-fidelity models and learn the actual traffic route network for better traffic flow management.

Ren and Li [195] compared flight trajectory data for China and US to learn the air traffic routes, and discovering airspace utilization patterns and some significant network structures. The horizontal efficiency of enroute flows was evaluated by Liu et al. [196], accomplished with the help of trajectory clustering for US airspace.

The authors estimated the horizontal efficiency by measuring the ratio between the length of the shortest routes and the real trajectory lengths. They also developed a statistical model to assess the impact of various parameters on trajectory inefficiency. It was observed in this study that the most significant causal factor for enroute inefficiencies is convective weather.

A number of clustering algorithms have been used in determining the spatial and temporal traffic patterns using flight data. The significant clustering methods that

are widely used include OPTICS [172], Hierarchical Density-Based Spatial Clustering of Applications with Noise (HDBSCAN) [197], k-means [198], and DBSCAN [173].

Similarly, Samantha et. al. [199] also contributed to making modern aviation systems safer by means of high-precision trajectory prediction and robust anomaly detection methods. This work also focused on the terminal airspace trajectory prediction and anomaly detection methods by the identification of air traffic flows. Since the air traffic flow patterns are identified using clustering algorithms whose performance depends on the characterization of an appropriate distance function, the selection of this distance function like Euclidean distance becomes a challenge due to the divergent and convergent nature of traffic airflows within the terminal airspace. The novelty of this work is the adoption of a weighted Euclidean distance function to improve trajectory clustering within the terminal airspace. In this work, numerous weighting schemes were evaluated by applying the HDBSCAN algorithm to cluster the trajectories. The key finding in this work was that if the trajectory points closer to the border of the terminal airspace, but not necessarily at the border, are weighted highest, then a more accurate clustering is computed. The above study determined air-traffic flows by identifying patterns in terminal maneuvering areas, and more emphasis was placed on optimizing a weighted distance function so that the best possible clustering is achieved. This study doesn’t address any UAV traffic flows in the terminal maneuvering areas within a densely populated low-altitude highly dynamic airspace.

In a PhD study [155], a flight trajectory data analytics framework was developed to provide a high-fidelity characterization of air traffic flows from large-scale aircraft-tracking data. This work applied machine learning methods to discover spatial and temporal trends in aircraft movement. The proposed framework allowed to automatically learn the airspace structure, assess the use of the airspace and identify patterns of airspace usage. For this, it included three modules: (1) clustering flight trajectories at the spatial scale to identify trajectory patterns, (2) trajectory classification by using Random Forests algorithm to assess flight trajectory conformance against the learned airspace structure and identify air traffic

flows, (3) clustering air traffic flows at the temporal scale to identify traffic flow patterns. This framework was used to obtain a detailed characterization of air traffic flows in the terminal areas of multi-airport systems including New York, Hong Kong and Sao Paulo. This work also highlighted the influence of weather conditions on airspace use and traffic performance and recommended the integration of weather forecasting in decision-making. It utilized the above data analytics conclusions to develop a data-driven approach for airport capacity planning towards an improved decision support system. This decision support system used predictive modelling for capacity estimation and prescriptive modelling for capacity allocations. The authors used a random forest–based supervised learning method to predict traffic flow patterns.

Although the above PhD thesis covered a lot of facets in air traffic flow prediction, it did not evaluate the enroute traffic behaviors and thus did not visualize the routing patterns between origin and destinations. Moreover, the impact of adverse weather conditions on the discovery of feasible rerouting options was not addressed. Furthermore, the work did not cover the domain of TMA or enroute traffic patterns in the UTM context.

A novel framework for analyzing air traffic flow was proposed in [156]. This framework is based on hierarchical density-based spatial clustering of applications with noise (HDBSCAN), an improved version of the clustering algorithm DBSCAN. A single input parameter is used to manage clusters of distinctive densities in this improved version. In this study, two methods – symmetrized segment-path distance (SSPD) and Euclidean distance (ED) – that are based on two different distance functions were evaluated. Traffic patterns in both terminal and enroute areas are widely analyzed using this framework; thus, it is a quite useful framework. Although this work employed a new dynamic clustering algorithm for predicting air traffic flows, the focus is on the distance function selection for better optimization results. HDBSCAN may be utilized for clustering UTM data traffic flow to consider the effects of noise.

Air traffic trajectories can be separated by an approach used in [157], carried out in a constrained area through operational procedures. In the Toulouse terminal

maneuvering area, the clustering algorithm is applied on a set of real trajectories. It is observed that a better understanding of traffic structure can be achieved by a DBSCAN-based clustering method, and schedule landings of aircraft can be controlled at Toulouse–Blagnac Airport. A cluster of significant trajectories with useful information has also been identified through this study. Moreover, these trajectories are paving the way for a probabilistic approach to carry out risk assessment in air traffic. Although this work validated the use of DBSCAN-based data analytics methodology for air traffic flow separations, it did not disclose any specific patterns' classification and the scope of terminal maneuvering areas in ATM space.

The scope of analysis in all previously mentioned studies is constrained to air traffic management and does not take into account the challenges in the UTM context. Furthermore, in operational performance analysis, the influence of airspace structure is generally disregarded in older works. Moreover, the effect of weather conditions in defining air traffic flows has been examined in a few studies. To compare and characterize the air traffic performance at various scales, the entire flight trajectory has been considered in this work, in contrast with previous works, and includes terminal maneuvering areas and enroute trajectories for UTM airspace.

3.3 Methodology

3.3.1 Simulation Setup

The suggested methodology is evaluated and verified by running simulations for the airspace of Bedfordshire, UK. Figure 3-1 depicts places that may be restricted for flights, such as airfields, recreational areas and prison. They include four airfields: Luton, Cranfield, Halton and Old Warren (Orange), four recreational areas: Dunstable, Sandy, Cardington and Graveley (Yellow), and Milton Keynes Prison (Blue). The blue color trajectory corresponds to Covid-19 sample missions by UAVs (1,2,3), the magenta color trajectory shapes represent the package delivery service by UAV-4. The cyan color trajectory represents the package delivery service by UAV-5. The red color trajectory presents the fire surveillance missions, and the black lines belong to recreational users.

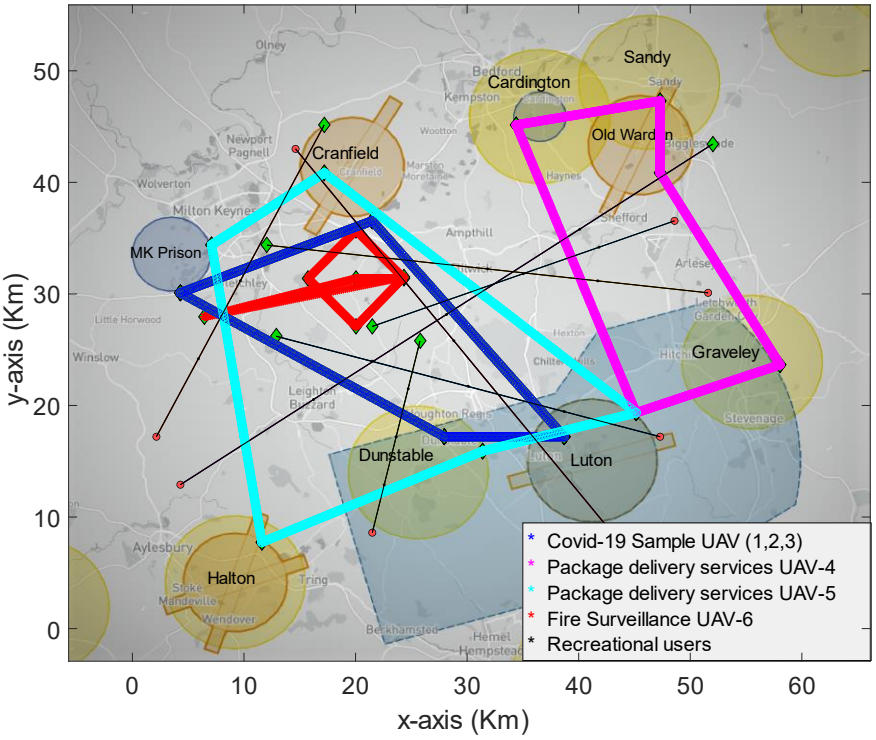

Figure 3-1 Simulation environment created by Air Map

To make the simulation more realistic, the following simulation parameters are taken into consideration:

1) Structure of airspace: The simulation framework utilized in this study does not support the simulation of ascending and descending maneuvers, resulting in the display of UAVs' cruise phase in a 2D format (x, y).
2) Fixed start and end position**:** As part of the simulation, the UAV-mission start and end positions have been set to mimic typical everyday operations and emergency services.
3) Priority levels**:** Due to the urgency of some flights, it is necessary to prioritize these over less pressing services [55][56]. In this study, each flight has been assigned a service priority level, ranging from level 1 (highest priority) to level 4 (lowest priority). The following is a description of each level of priority:
    a) Fire surveillance services using loiters' trajectory patterns.
    b) COVID-19 sample test delivery service to multiple clinics.
    c) Package delivery service from multiple post offices.
    d) Recreational UAV user (hobbyist flyers).

4) Dynamic no fly zones (NFZs): In some hours, airfields and recreation areas considered for simulation are dynamic in nature. This dynamism is random, making some areas available and others unavailable during the whole hour.
5) Random departure time**:** To make the system more realistic, the exact time at which each hobbyist's UAV departs is set randomly between 1 and 10 minutes in a 1-hour simulation scenario period.
6) Weather ambiguity: Various weather conditions, categorized as adverse and severe, have been taken into account.
7) Throughout this study, the assumption has been made that communication blackouts do not occur and there is consistent communication performance across all regions.
8) As discussed in the earlier section (2.4.1), this study employs particle swarm optimization to determine optimal paths for UAVs, guiding them from the starting point of the service to the delivery destination. The algorithm incorporates a least mean square cost function that calculates the sum of Euclidean distances between the UAV and all obstacles to avoid collisions. The cost function includes a coefficient β that balances particle fit, global fit, and obstacle avoidance to generate an optimal trajectory. Higher β values prioritize obstacle avoidance but may lead to larger deviations and sharper angles, while lower β values reduce deviations and sharp angles at the expense of obstacle avoidance, as observed in the simulation scenario figures. The study places a strong emphasis on safety by prioritizing obstacle avoidance, even if it results in some sharp angles. The particle dynamics, represented by the inertia weight damping coefficient (wdamp), are considered in PSO's velocity calculations. However, the study does not incorporate the UAV's pitch, yaw, and roll dynamics when generating the trajectory.
9) Deconfliction Strategy: To resolve potential conflicts among UAVs, the author of this thesis [200] proposed a set of deconfliction strategies. These strategies were extensively discussed in section 2.5 and subsequently employed for the purpose of deconfliction.

3.3.2 Weather Effects' Implementation

For a UAV's operation, weather hazards are classified into moderate, adverse and severe [201]. Moderate hazards are those that result from phenomena that reduce visibility but do not harm the aircraft. Severe weather hazards are those that might cause significant damage to an aircraft or result in its loss of control. Adverse hazards weather conditions that have potential to cause loss of control, loss of communication and diminished aerodynamic performance and may negatively affect the operator. Thus, it is well accepted that flying in severe weather should be avoided.

Drones can be blown off course by strong winds, making them impossible to control during takeoff, in-flight, or landing, thus resulting in a crash. When the wind direction is normal to the direction of the UAV, the wind forces the UAV to change its trajectory [202]. Strong winds have the potential to affect an unmanned aircraft's ground speed and flight path. Wind speeds can easily exceed the maximum speeds of UAVs, unlike manned aircraft. Fixed-wing UAVs have an aerodynamic structure that helps them maneuver through air streamlines, whereas even at lower wind speeds, rotary-wing UAVs are more likely to struggle. That is why fixed-wing UAVs have a higher maximum speed than rotary-wing UAVs [201]. The mathematical representation of high-speed winds' impact on the operation of UAV is presented as follows:

Suppose the direction of wind is denoted by β_1, and the deviation in a UAV's trajectory caused by the wind is denoted by β_2. m_1 and m_2 are vertical line equations slope. The relationship between the slopes and directions of wind and the deviation of the UAV is as follows:

$$m_1 = tan\, \beta_1 \tag{3-1}$$

And

$$m_2 = \frac{y(z+1) - y(z)}{x(z+1) - x(z)} \tag{3-2}$$

where z is the simulation discretized steps. A vertical line is formed at the instant the UAV interacts with the wind at every step. The change in direction along x-axis and y-axis is defined by:

$$x = cos\,\beta_2, \tag{3-3}$$

$$y = sin\,\beta_2$$

$$Tan\,\beta_2 = \frac{(m_2 - m_1)}{\left(1 + (m_2)(m_1)\right)} \tag{3-4}$$

Table 3-1 shows two different weather classifications based on the Beaufort wind scale and rainfall precipitation.

Table 3-1 The scale of weather classification used in this study

Weather types	Adverse	Severe
Wind (m/s)	13–23	> 33

3.3.3 Framework For Characterization of Air Traffic Pattern

For the identification and characterization of UAV traffic flow patterns, a flight trajectory–based framework that uses the DBSCAN algorithm, and the Gaussian kernel density estimator has been described here. Since the UTM traffic flows are related to UAV missions following the same trajectory paths on different time schedules, deviations in the planned paths are expected due to the presence of dynamic airspace architectures, weather uncertainties and random hobbyist flights. The characterization of traffic flows thus needs the detection of trajectory patterns for each mission separately for characterization. In the past, a number of researchers have conducted ATM traffic-flow pattern identifications, using DBSCAN for terminal-maneuvering areas around airports [153]–[157] as already discussed in Section 3.2. The proposed framework comprises the following functions:

1) DBSCAN–based clustering of UAV trajectories in the spatial domain.
2) Identification of UAV mission-based clustered groups.
3) Statistical analysis of noisy cluster groups and valid mission trajectories.
    a) Valid clusters are clusters that belong to the class of one of the three planned missions.
    b) Noisy clusters are clusters that pertain to random flying of UAVs by hobbyists.
4) Characterization of UTM traffic flows using detected mission clusters.
5) Congestion analysis of UTM airspace Using a Gaussian kernel density estimator for UAV trajectories.

This framework uses the above functions to automatically learn and identify the airspace structure. The main purpose of trajectory clustering is to find groups of trajectories in the spatial dimension that belong to valid missions in the presence of random and dynamic factors. The framework also helps to find congestion patterns in the UTM airspace through calculations of air traffic density. The methodology of the above-stated flight trajectory framework for the characterization of air traffic flow for UTM is shown in Figure 3-2.

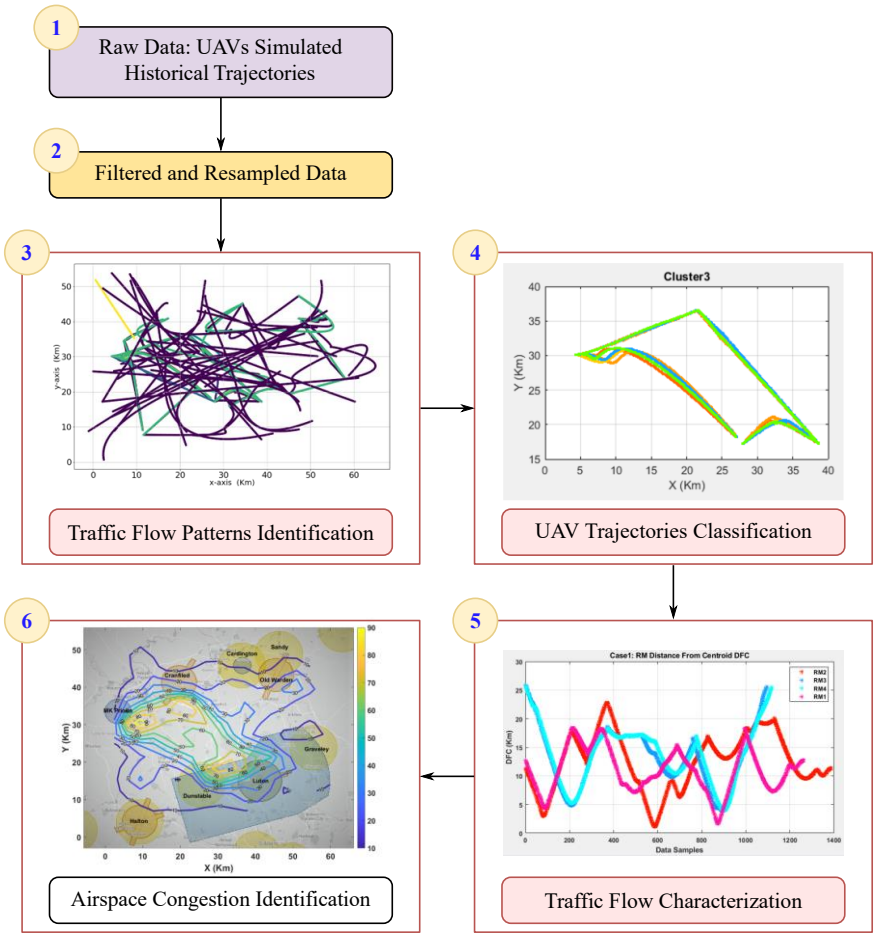

Figure 3-2 Overview of the UTM airspace characterization approach

Clustering tasks can be performed using a variety of approaches and algorithms, which can be classified into three sub-categories: partition-based (k-means), hierarchical (divisive, agglomerative) and density-based (DBSCAN). With normal-shaped clusters, hierarchical- and partition-based clustering techniques are extremely helpful. DBSCAN, on the other hand, is more efficient in detecting outliers or arbitrarily shaped clusters. In the presence of abnormal trajectory profiles, DBSCAN identifies the core trajectory patterns. It has the ability to identify non-convex clusters and eliminate the need to predetermine the number of clusters.

In this study, DBSCAN is chosen because of two reasons: (a) the algorithm can decide the number of clusters automatically, and (b) outliers can be identified. These characteristics meet the requirement of trajectory clustering because (1) the number of typical UAV air traffic flow patterns is unknown with random factors

such as dynamic objects, wind and random flights of UAV hobbyists, (2) UAV flights undergoing abnormal deviation can also be extracted from clusters as outliers. Successful applications of DBSCAN in air transport research can be found in the literature [203], [195]. DBSCAN clustering depends on the input parameters epsilon and minPts [155] as follows:

1) Epsilon: Distance threshold. It specifies neighbor distance. If the distance between two points is less than or equal to an epsilon, then it is considered neighbor.
2) Minimum points (minPts): For defining cluster, minimum number of data points.

Core points, border points and outliers are defined based on the above parameters:

1) Core point: A point is considered core point if it is surrounded by at least minPts points (including the point itself) within the epsilon radius.
2) Border point: It is part of a cluster because it is within the epsilon of the core points, but it doesn't meet the criterion of minPts.
3) Outlier: When the point is not a core point and is also unreachable from any core point, then it is called an outlier.

The clustering process works on three main concepts, namely epsilon neighborhood, density reachability, and density connectivity, given below:

The D_i observation epsilon neighborhood contains all observations that are within the epsilon distance:

$$N_\varepsilon\,(D_i) = \{D_J \in Z / d(D_i, D_j) \leq \varepsilon\; E_d(D_i, D_j) = \|D_i - D_j\|_2\} \quad \textbf{(3-5)}$$

Where D_i and D_j are two different trajectories, and $\boldsymbol{\varepsilon}$ represents epsilon. The observation trajectory D_J is the density reachable from D_i if

$$D_J \in N_\varepsilon\,(D_i) \quad \textbf{(3-6)}$$
$$|N_\varepsilon\,(D_i)| \geq minPts$$

If there is chain of $D_i, \ldots, D_J$ such that each subsequent observation is directly density-reachable from the preceding one, then D_J is density-reachable from D_i. D_J is the density connected to observation D_i if both observations are density-reachable from another observation D_k.

The algorithm begins with an arbitrary database instance and determines its epsilon neighborhood. Observation and neighborhood start clustering if it contains at least minPts, which is the core-point condition. The same procedure is applied for the neighbor's ε −neighborhood iteratively retrieved until the border points are achieved, or else the observation is classified as noise, and the algorithm moves on to the next database instance. If there are newly discovered core points, then this observation instantly labelled as noise can become part of the cluster, and the points are defined as density-connected points. Figure 3-3 illustrates the DBSCAN concept.

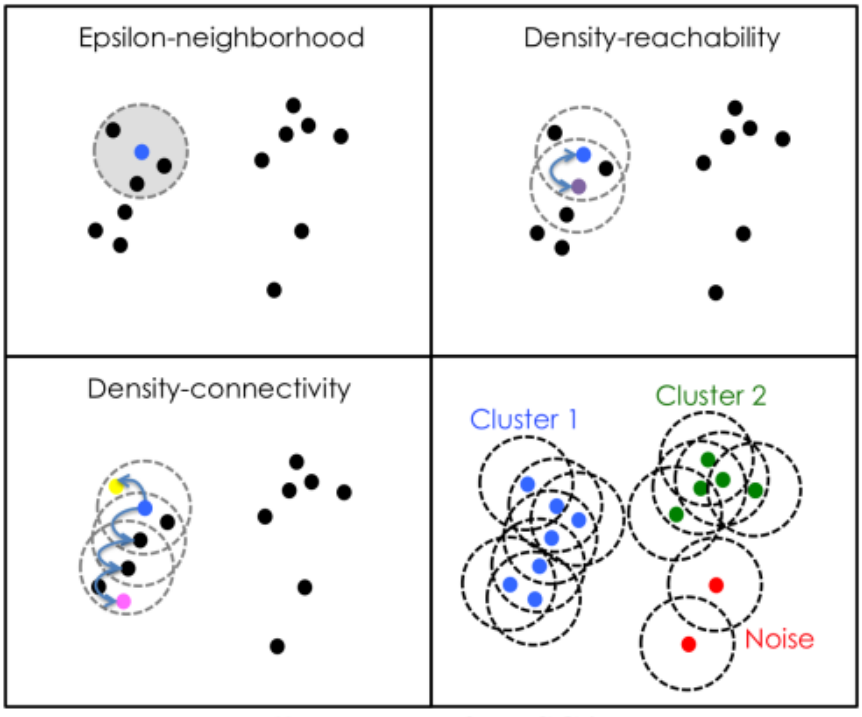

Figure 3-3 DBSCAN clustering [155]

The following methodology is adopted in this regard:

1) Pre-processing data by resampling the data for better resolution.
2) Tuning the DBSCAN parameters heuristically for extracting valid mission trajectories as one cluster or class.

3) Labelling different cluster groups pertaining to one mission.
4) Visualizing the group of mission clusters as aggregate or combined plots for identifying different deviation patterns for each mission.
5) Reporting the statistics of trajectory points per cluster.
6) Calculating the percentages of valid and noisy clusters.

3.4 Simulation Scenario and Results

The absence of historical UTM traffic data necessitates the generation of data using a simulation framework to analyze UAV traffic-flow patterns. Various scenarios were considered to illustrate the problem, including COVID-19 sample delivery to multiple clinics, package delivery with square loitering-based fire surveillance missions, and the presence of flying hobbyists operating their UAVs simultaneously.

It is assumed that the Luton–Bedfordshire National Health Service (NHS) hospital is currently housing a fleet of three UAVs (UAV1, UAV2, and UAV3) for COVID-19 sample test delivery missions to multiple clinics in a defined distribution network. The UAVs' shuttle service will occur each hour. The delivery of mail packages is from Luton Central Delivery Office to the multiple-post office. This mission fleet consists of two UAVs (UAV4, UAV5); one UAV will be allocated for the areas of Graveley, Old Warren, Sandy and Cardington, the second UAV will cover the remaining areas (Cranfield, Halton, Dunstable, and Milton Keynes Prison. The purpose of the central mail office to use two UAVs for delivering packages to multiple post offices is to reduce delivery time and gain customer satisfaction.

The firefighting-based rescue operations rely heavily on the surveillance operation using cameras. The initial videos and pictures taken by the payload cameras would help define the course of action in fire-fighting operations. Currently, only one square loiters operation is considered in our scenario. We are considering a UAV-6 available at Milton Keynes Prison Fire Station as a base station for these missions. Additionally, the scenario encompasses the inclusion of hobbyists flying their UAVs alongside the aforementioned planned missions. It

is assumed that these flying trajectories consist of a single leg with random start and finish locations. The plan and schedule for each mission are presented in Table 3-2 below.

Table 3-2 The description and schedule for UAVs missions

Missions	Mission 1 COVID delivery sample test	Mission 2 Packages delivery	Mission 3 Fire Fighting (Surveillance)	Mission 4 UAV flying by hobbyists
Trajectories number	36	30	6	28
Priority	2	3	1	4

Furthermore, the assumption is made that the UAVs are readily available at the start of the scenario and are flying at a consistent speed of 90 km/hr . It is further supposed that the times of delivery depend on the sequence of customers to be served along flight routes. Breakdown and maintenance times and costs are not considered. Each UAV follows a particular route, starting and ending at the same depot; the mission of a UAV is completed when it completely follows the assigned route and reaches back to the depot safely. To simulate a dynamic airspace with increased complexity, a total of 100 UAV trajectories were considered. Among these, 73 trajectories were allocated for special UAV missions, while the remaining 27 trajectories were designated for random flights conducted by hobbyists . Technical parameters of the UAVs are presented in Table 3-3.

Table 3-3 Technical parameters of UAVs

Parameters of UAVs	Value	Unit
UAV type	Rotary-wing	--
Flight time	30	minutes
Cruise Speed	90	km/h
Wind resistance	10	m/s

A multi-mission scenario was simulated for the Bedfordshire area, spanning from 9:00 am to 12:00 pm. The simulation involved the creation of three distinct sub-

scenarios corresponding to different time intervals within this timeframe, which are described as follows:

3.4.1 Scenario 1 – NFZ With no Weather Constraint and Dynamic Obstacles

This simulation runs between 9:00 am and 10:00 am in the environment illustrated in Figure 3-4. In this simulated scenario, all nine NFZs are static with no dynamic obstacles and no weather constraints. As a result, no UAV could fly over four airfields, four recreational areas and one prison. However, any missions related delivery can be accomplished for these regions if applicable. The recreational users, however, are prohibited to fly over these NFZs.

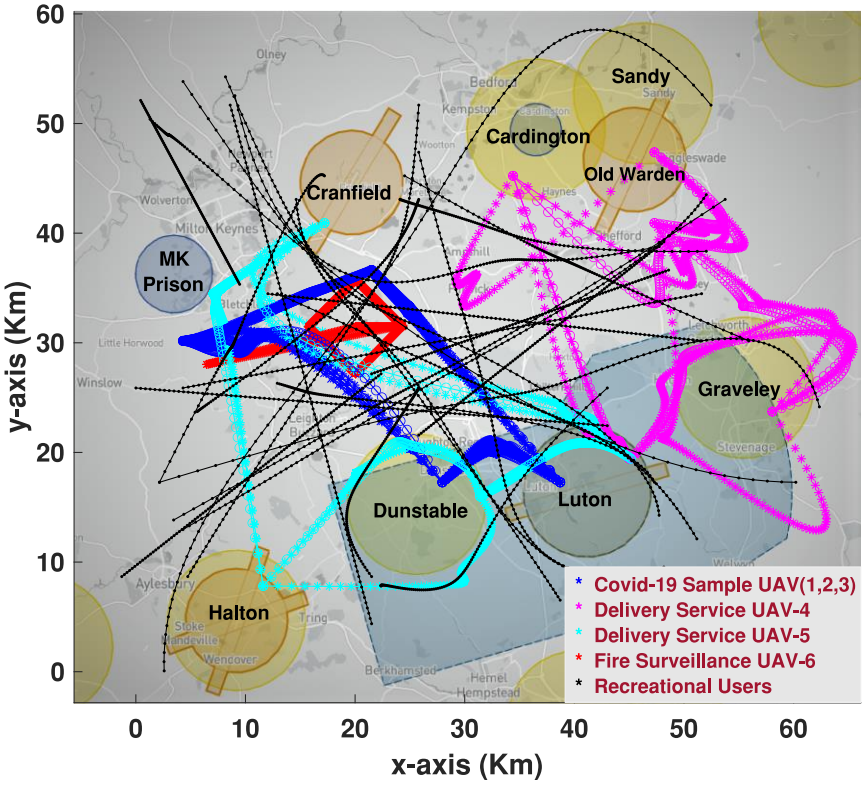

Figure 3-4 Simulation scenario 9:00 am to 10:00 am

3.4.2 Scenario 2 – Static Airfield and Dynamic Recreational Area with Extreme Weather Conditions

The second simulation scenario runs between 10:00 am and 11:00 am. The difference with the first scenario is that any NFZ corresponding with recreational

areas are dynamic, making some areas available and others unavailable in this hour. Figure 3-5 shows Dunstable, Sandy, Cardington and Graveley opened their airspace. Therefore, some UAVs could fly over dynamic NFZs during the simulation time. This scenario also incorporated the effects of extreme weather, dynamic in spatio-temporal axes, which would result in severe damage or loss of control of an UAV. Changes in extreme weather condition make it practically impossible to finish the flight of UAVs and have a significant impact on the success of planned missions of surveillance and COVID-19 sample distribution. Thus, UAVs avoid flying in extreme weather.

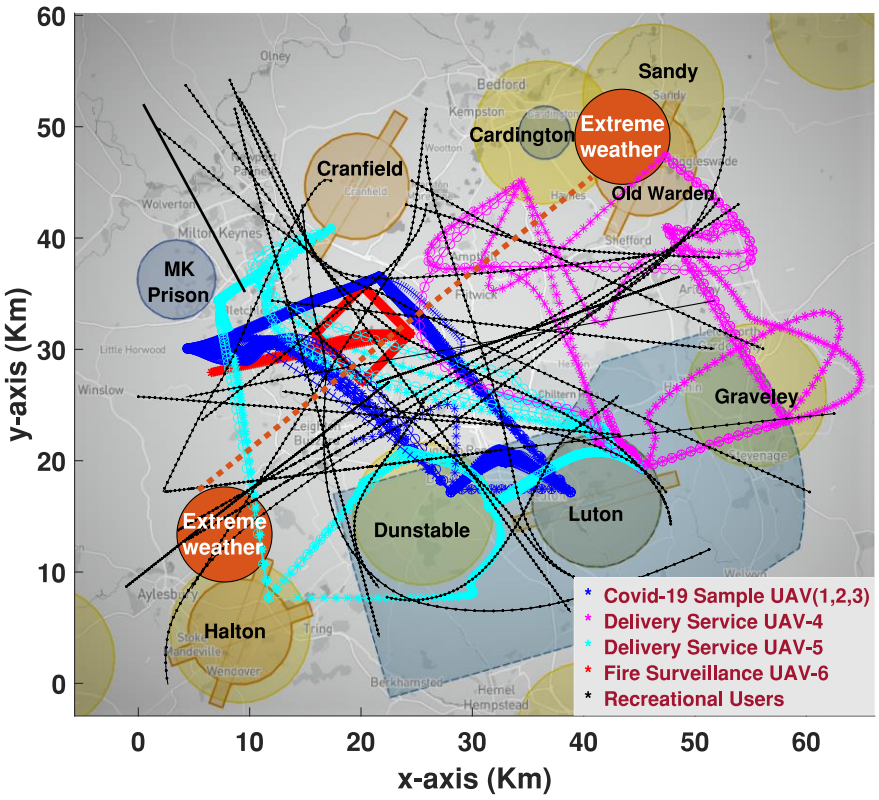

Figure 3-5 Simulation scenario 10:00 am to 11:00 am

3.4.3 Scenario 3 – Dynamic Airfield and Static Recreational Area with Different Weather Condition (Rain and Wind)

In this simulated scenario, airfields are dynamic, while all four recreational areas and the prison are kept static. Among the four airfields, Luton and Cranfield airfields are available, as shown, and therefore UAVs of flying hobbyists can fly over Luton and Cranfield areas at some points. This scenario takes place

between 11:00 am and 12:00 pm. Effects of adverse rain and wind have been considered, as shown in Figure 3-6.

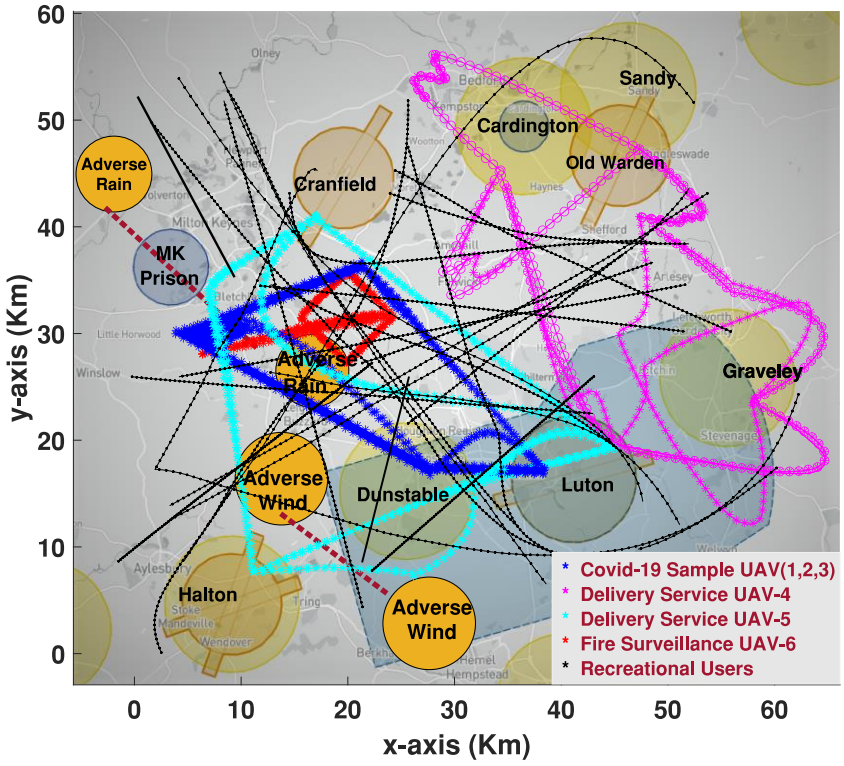

Figure 3-6 Simulation scenario 11:00 am to 12:00 pm

Since the data is generated using the PSO simulation framework, no data filtering is required; however, for the sake of better resolution, the trajectory data for 100 UAVs with 100 interpolated sample points is up-sampled to about more than 20000 sample points for smooth DBSCAN clustering.

The DBSCAN algorithm is applied for the clustering and classification of the above mixed-mission data for three scenarios; further details are provided in the following section. Google Colab–based Python environment is used to run the DBSCAN clustering algorithm. The parameters epsilon ($\boldsymbol{\varepsilon}$) and minimum points (**minPts**) are tuned for each dataset accordingly, in order to acquire better results. In this manner, different mission-trajectory patterns are identified in this unsupervised learning mechanism. Additionally, the hobbyists' single-leg trajectories were identified as outliers in this work. The clustering for these scenarios is tuned and

achieved with different epsilon ($\boldsymbol{\varepsilon}$) and minimum points (minPts) using the heuristic method are presented in Table 3-4.

Once the clustering is done using the DBSCAN algorithm, a detailed cluster analysis is conducted using the trajectory matrix and labelled clusters to visualize different mission trajectory patterns and their shapes. Furthermore, aggregate or combined cluster patterns for different missions are plotted and analyzed. The analysis is also supported with the help of the percentage of valid and noisy clusters found using DBSCAN. The tally of different trajectory patterns of various missions is also obtained. We further evaluated the percentages of various valid mission trajectories detected per cluster in this analysis using pie charts.

Table 3-4 DBSCAN cluster's parameters for simulation scenarios

Scenario	DBSCAN Parameters		Total Clusters	Valid Mission	Noisy Clusters
	eps	minPts			
0	0.6	3	11	9	2
1	0.6	3	9	7	2
2	0.6	3	7	5	2
3	0.4	3	12	7	5

3.5 Airspace Traffic Flow Patterns, Characterization and Density discussion

3.5.1 Identification of UTM Air Traffic Patterns

This section defines the results of identifying different trajectory patterns in the planned UAV missions explained in section 3.4 above. The mission legs and their trajectories are shown in Figure 3-4,Figure 3-5 and Figure 3-6 under three different scenarios. These three scenarios perturb the ideal mission plans with the introduction of various dynamic random factors such as recreational areas, airfields and different uncertain weather conditions that cause trajectory

deviations and result in different mission trajectory patterns that can't be identified visually.

These scenarios run for each hour of the day, ranging from Scenario-1 (9:00 to 10:00 am) to Scenario-3 (11:00 am to 12:00 pm). Variations in airspace structures and weather conditions during these times were considered. The deliberate choice of a one-hour temporal time horizon aims to enhance the precision of the analysis, revealing nuances in traffic behavior across distinct hourly periods and enabling a more insightful exploration of temporal dynamics within the clustered data. However, it is important to note that this study did not consider clustering analysis at the temporal scale. Instead, it focused on utilizing a trajectory clustering scheme to identify spatial trajectory patterns, which define the as-flown route structure in the airspace of interest. Additionally, a trajectory classification scheme was employed to assess flight trajectory conformance to the learned airspace structure and identify air traffic flows.

Initially, an ideal scenario was simulated to establish a reference UAV mission trajectory pattern, representing simplified trajectory geometries within an open airspace devoid of static or dynamic obstacles and weather constraints. This ideal scenario served as a crucial step in validating the clustering methodology. Subsequently, the methodology underwent rigorous testing in more complex scenarios (Scenario-1-3), incorporating No-Fly Zones (NFZ), dynamic obstacles, and weather constraints. Within these scenarios, the methodology effectively handled missions characterized by irregular shapes. The shapes and trajectories of these missions were not predetermined but influenced by factors like wind patterns and dynamic obstacles, resulting in varying trajectory shapes. Despite these significant challenges, the methodology successfully identified and clustered these missions based on their distinctive characteristics. The subsequent sections provide a detailed discussion of the results and analysis for both the ideal case scenario and the three main scenarios.

3.5.1.1 Scenario 0 – Ideal Case without NFZ, Weather Constraints and Dynamic Obstacles

The intent of the DBSCAN clustering in this work is to assign trajectories to clusters in an unsupervised way similar to [154], [158], thus identifying and

separating the regular mission trajectories within the noise of recreational hobbyist missions. The input features used in DBSCAN-based trajectories clustering are 2D horizontal position vectors (X, Y) of all UAVs. These feature data are resampled to increase the resolution and provide equal spacing between consecutive trajectory points of each UAV. The proposed clustering methodology is validated using simple geometry. In this validation process, UAV missions are represented as regular polygons and conducted in an airspace that lacks static, dynamic obstacles, or weather constraints. Examples of such missions include straight-line polygons and square parallel line missions, as illustrated in the figure below.

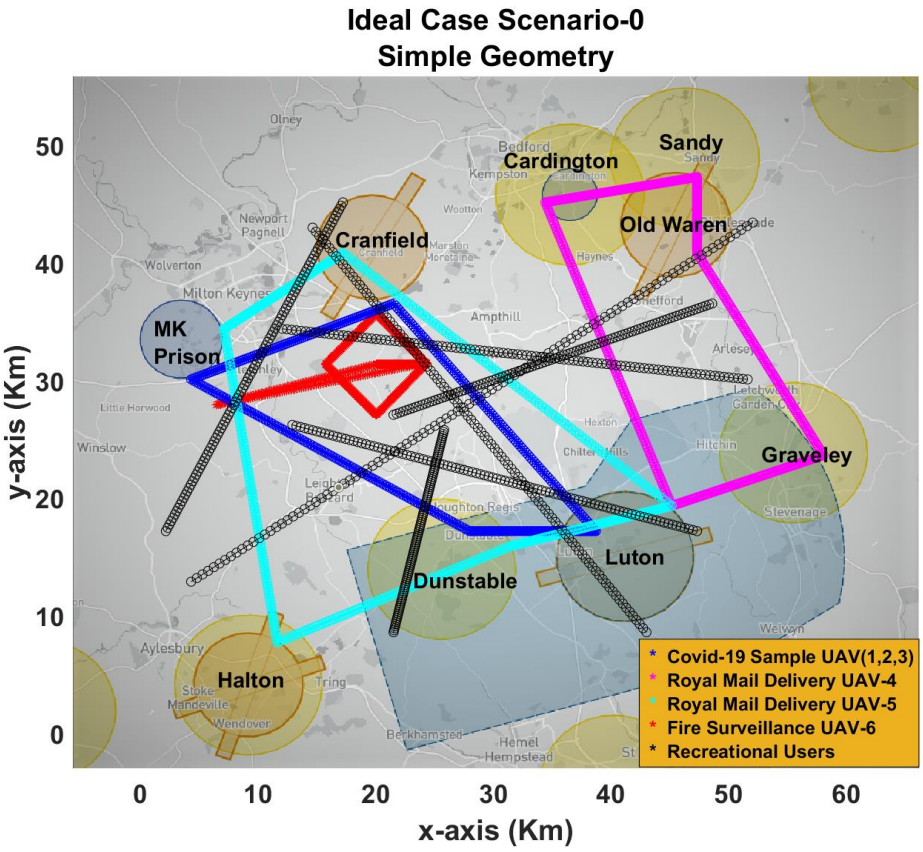

Figure 3-7 Simulation-scenario environment for the ideal case scenario

The DBSCAN has successfully detected regular missions from the provided trajectory data set of the whole airspace without prior knowledge of any mission. The clustering for this scenario is tuned and achieved with epsilon ($\boldsymbol{\varepsilon}$ = 0.6) and

minimum points (minPts = 3). The detailed low-level clustered trajectory patterns detected with the above classifiers are depicted in Figure 3-8.

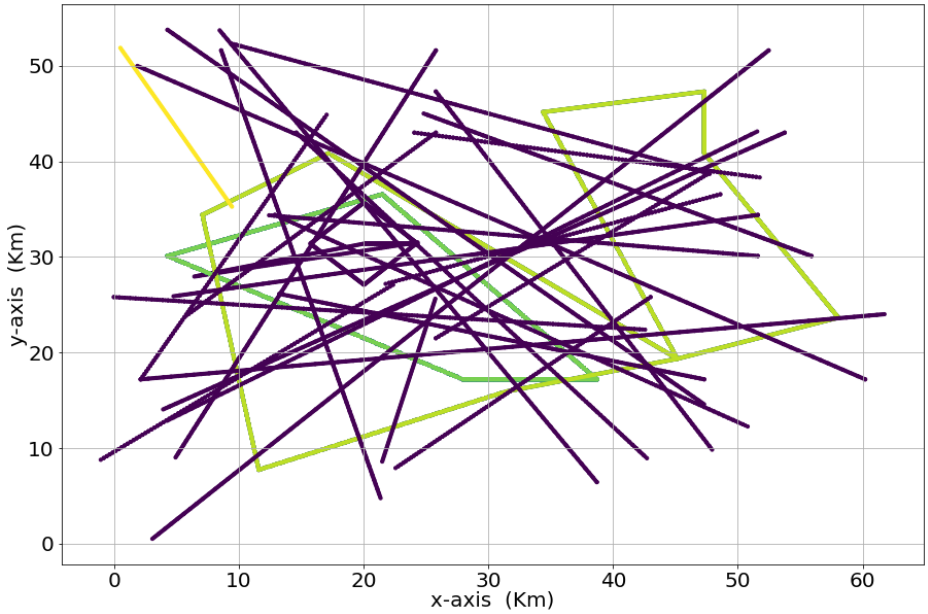

Figure 3-8 Scenario-0 DBSCAN results

It can be observed from the separated cluster graphs in Figure 3-9 that COVID-19 shuttle service trajectory patterns (UAV1, UAV2, UAV3) are detected with much higher numbers in clusters 3, 4, 5, 6, 8, 9, and 10. Also, the UAV4 and UAV5 package delivery service trajectory patterns are well detected and separated (clusters 7,11) from the denser areas where the outliers (such as random UAVs flying) have covered the airspace more comprehensively. Two types of trapezoidal trajectory patterns were detected, as shown in Figure 3-10.

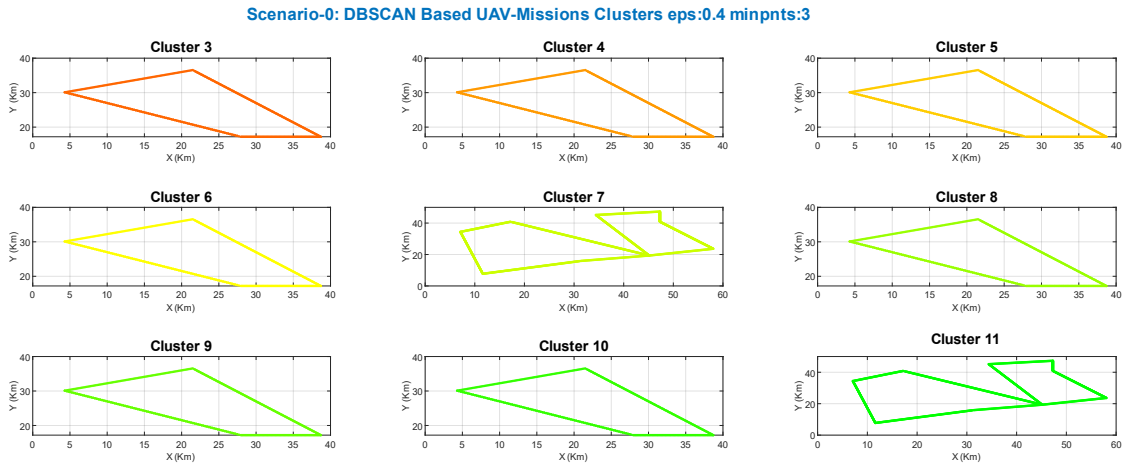

Figure 3-9 Scenario-0 DBSCAN detected mission cluster plots

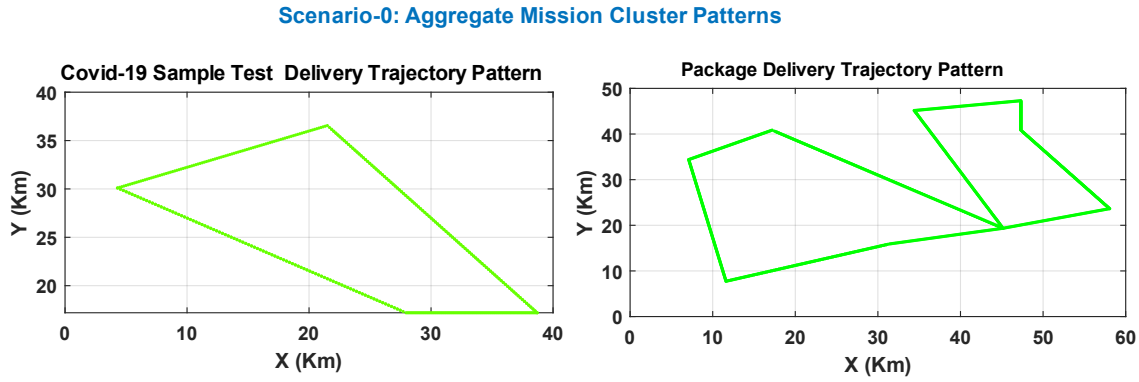

Figure 3-10 Scenario-0 Detected combined mission patterns

The percentages of trajectory points per cluster for Scenario-0 were evaluated and depicted in Figure 3-11 for further analysis. It is evident that most of the trajectories detected are of the COVID-19 sample delivery missions as shuttle service with 56% weightage compared with delivery package services that constitute 44% of the trajectories of missions.

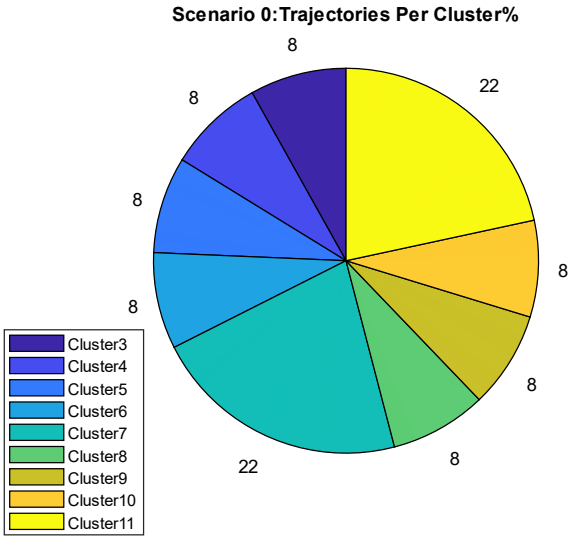

Figure 3-11 Scenario-0 Trajectory count per cluster (%)

3.5.1.2 Scenario 1 – NFZ with no Weather Constraints and no Dynamic Obstacles

The detailed low-level clustered trajectory patterns detected with the above classifiers are shown in Figure 3-12.

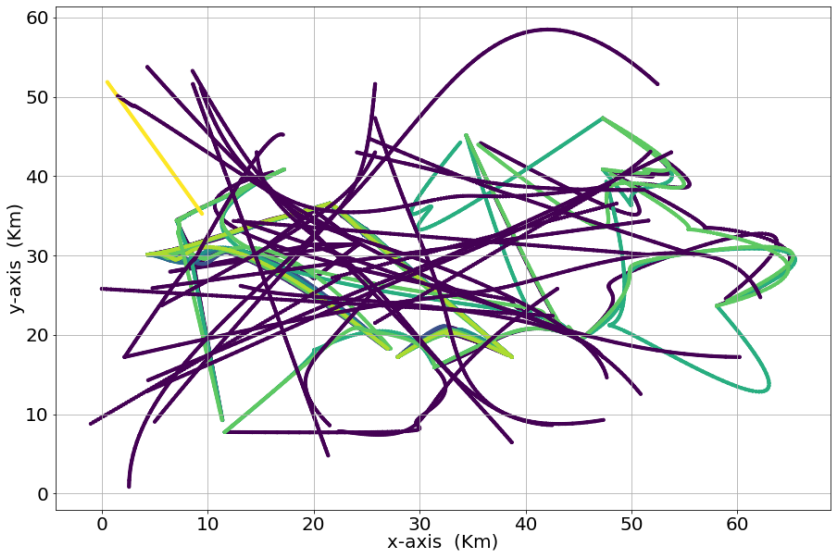

Figure 3-12 Scenario-1 DBSCAN results

It can be observed from the separated cluster graphs (Figure 3-13) that UAV1, UAV2, and UAV3 COVID-19 shuttle service trajectory patterns are detected with

a much higher confidence level in clusters 3, 4, 5, 6 and 9. UAV4 and UAV5 package delivery service trajectory patterns are detected in clusters 7 and 8.

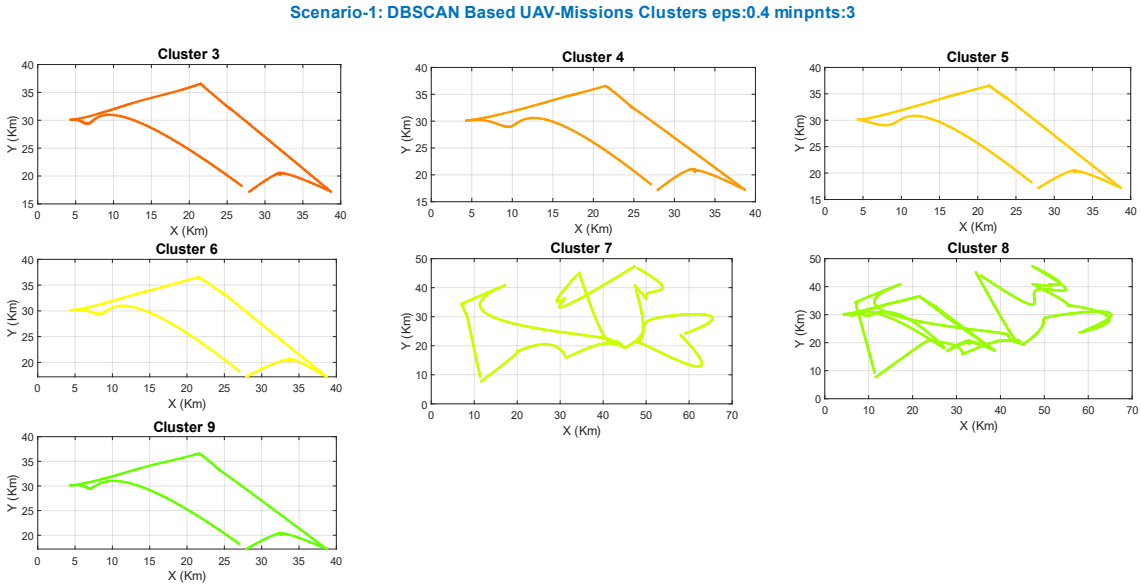

Figure 3-13 Scenario-1 DBSCAN detected mission cluster plots

As observed, the presence of active static NFZs has clearly deviated the ideal straight line–based trapezoidal trajectories to a distorted shape for UAVs for COVID-19 samples. Also, the package delivery service UAV trajectories are much distorted from the ideal shapes for both UAV4 and UAV5 missions. The combined trajectory patterns, as depicted with the help of DBSCAN, are also shown in Figure 3-14.

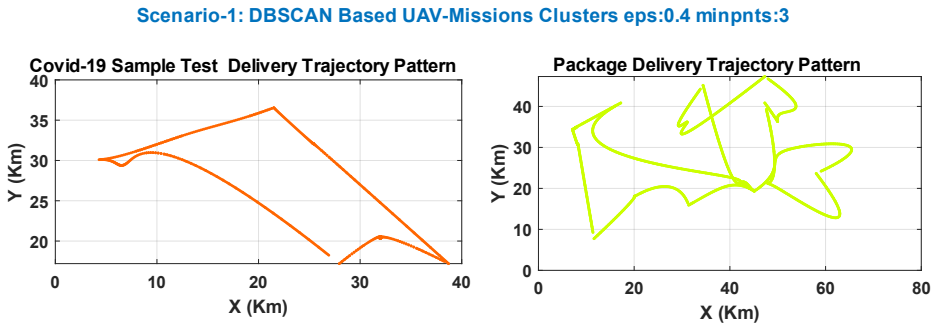

Figure 3-14 Scenario-1 Detected combined mission patterns

The percentages of trajectory points per cluster for Scenario-1 were evaluated and depicted in Figure 3-15 below. It is evident that most of the trajectories

detected belong to the COVID-19 sample delivery missions as shuttle service with 63% weight compared with delivery package services that constitute 37% of the detected missions.

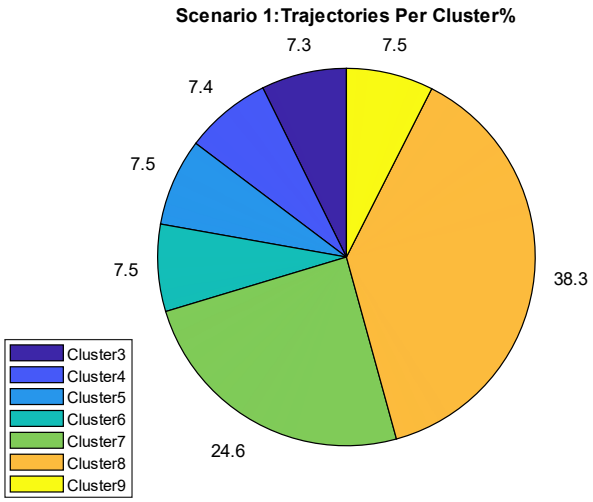

Figure 3-15 Scenario-1 Trajectory count per cluster (%)

3.5.1.3 Scenario 2 – Static Airfields and Dynamic Recreational Areas with Extreme Weather Conditions

The detailed low-level clustered trajectory patterns detected with the above classifiers are shown in Figure 3-16 and Figure 3-17.

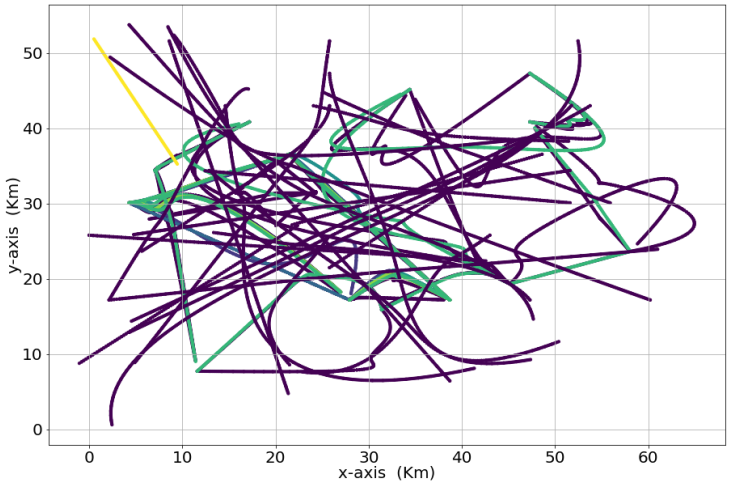

Figure 3-16 Scenario-2 DBSCAN results

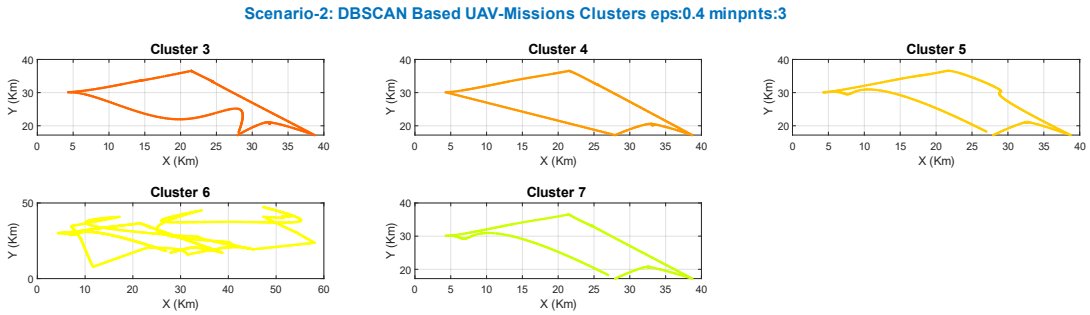

Figure 3-17 Scenario-2 DBSCAN detected mission cluster plots

It can be observed from the separated cluster graphs that UAV1, UAV2, UAV3's COVID-19 shuttle service trajectory patterns are detected with a much higher confidence in clusters 3, 4, 5 and 7. UAV4 and UAV5's delivery package service trajectory pattern is only detected in cluster 6. Moreover, the presence of active static NFZs, dynamic obstacles and extreme weather conditions has clearly deviated the ideal trapezoidal trajectories to much more distorted shapes for the COVID-19 sample UAVs. Also, it can be seen that the delivery package service UAVs' trajectories are much distorted from the ideal straight line–legs and Scenario-1 for both UAV4 and UAV5 missions. The combined trajectory patterns depicted with the help of DBSCAN are also presented in Figure 3-18. The percentages of trajectory points per cluster for Scenario-2 as represented in Figure 3-19.

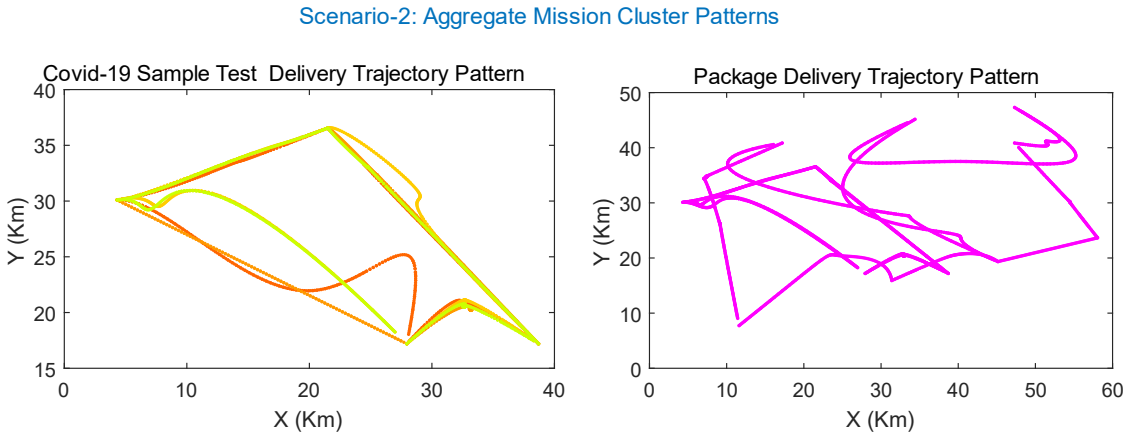

Figure 3-18 Scenario-2-Detected combined mission patterns

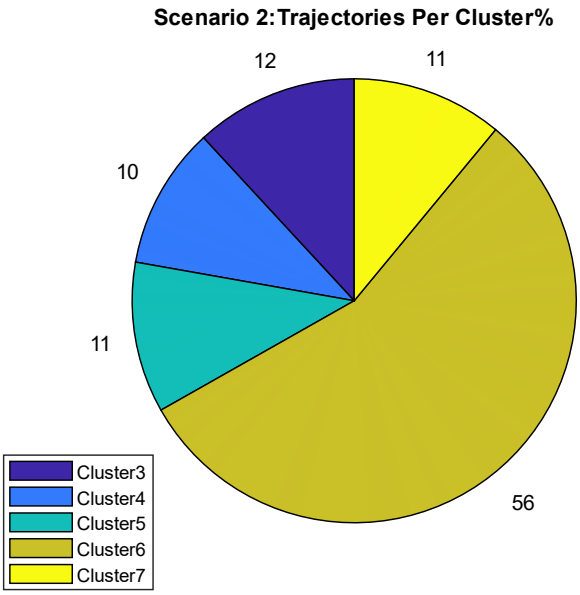

Figure 3-19 Scenario-2 trajectory count per cluster (%)

3.5.1.4 Scenario 3 – Dynamic Airfields and Static Recreational Areas with Different Weather Conditions (Rain and Wind)

The detailed low-level clustered trajectory patterns detected with the above classifiers is shown in Figure 3-20 and Figure 3-21.

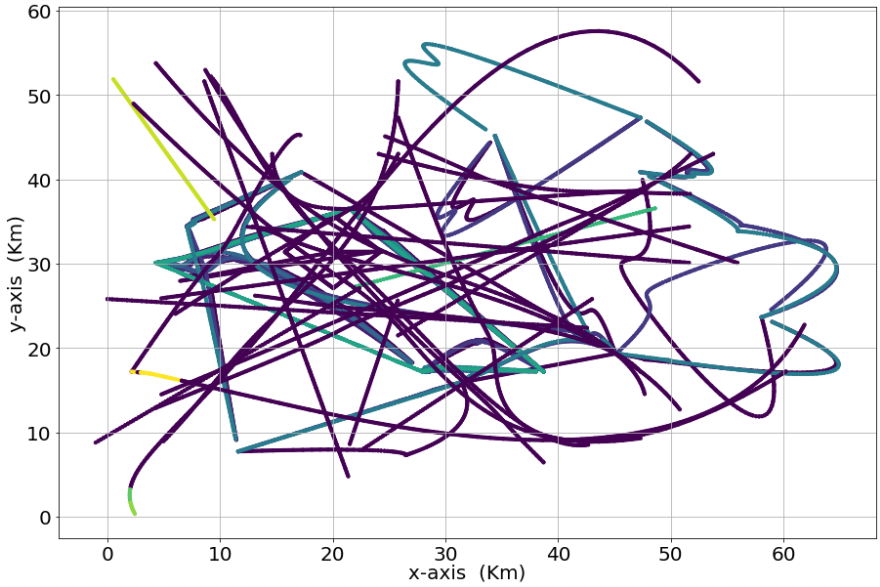

Figure 3-20 Scenario-3 DBSCAN results

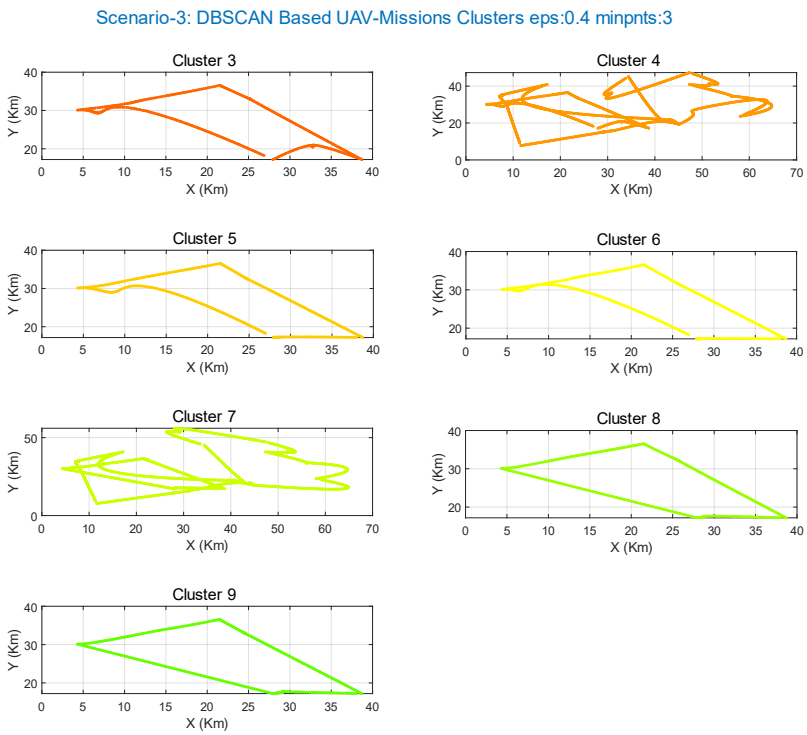

Figure 3-21 Scenario-3 DBSCAN detected mission cluster plots

It can also be noted that presence of dynamic airfields, adverse winds and rain conditions has clearly deviated the ideal trapezoidal trajectories to much more distorted shapes for the UAVs carrying COVID-19 sample. Additionally, the delivery service UAVs' trajectories are severely distorted from the ideal straight line–legs, Scenario-1 and Scenario-2 for both UAV4 and UAV5 missions. The combined trajectory patterns depicted with the help of DBSCAN are also presented in Figure 3-22.

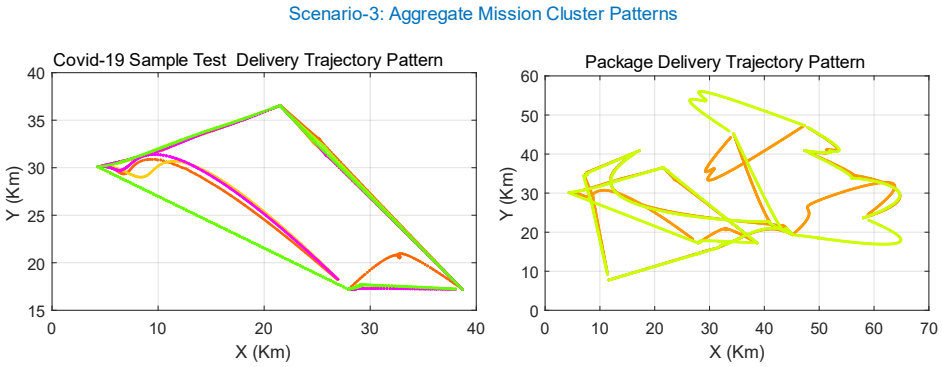

Figure 3-22 Scenario-3- Detected combined mission patterns

The percentages of trajectory points per cluster are represented in Figure 3-23.

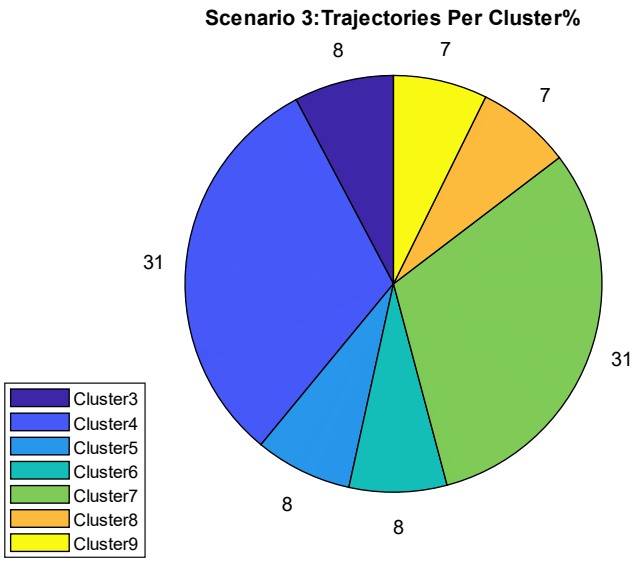

Figure 3-23 Scenario-3 Trajectory count per cluster (%)

3.5.2 Statistical Analysis of Clusters and Valid Trajectories

The performance of DBSCAN-based mission clusters' detection is depicted in Table 3-5 below. The table presents the percentages of valid mission clusters and noisy clusters detected. As observed, under ideal weather conditions and no obstacles, DBSCAN detected more than 81% of valid mission clusters from the provided dataset. It is also observable that in the presence of static NFZs, the deviation in trajectory patterns reduced detection capability to almost 77%. These percentage levels have been further reduced to 71% in the presence of static airfields and dynamic recreational areas along with dynamic extreme weather clouds that passed by in almost all of the missions' execution areas at different times during the hour, heading between south-west and north-east. This percentage of valid clusters' detection has been reduced to almost 58% with dynamic airfields, static recreational areas, and weather conditions like wind and rain.

Table 3-5 Valid mission & noisy clusters' percentages

Scenario	DBSCAN Parameters		Total Clusters	Valid Mission Clusters	Noisy Clusters	Valid Clusters %	Noisy Clusters %
	eps	minPts					
0	0.6	3	11	9	2	81.82	18.18
1	0.6	3	9	7	2	77.78	22.22
2	0.6	3	7	5	2	71.43	28.57
3	0.4	3	12	7	5	58.33	41.67

The analysis also encompassed the examination of the number of valid missions detected from the overall cluster groups that should belong to subsets of UAV missions as presented in Table 3-6. Shuttle service–type missions were identified in large numbers as compared with one-time fire surveillance missions. This may be due to the reason that fire surveillance square loiter is missed in the detection mechanism because of the interference of much higher hobbyists' UAVs single-leg trajectories in this area. This also indicated the presence of much denser trajectories from diverse missions, dynamic environments, and random UAV flights in this area.

Table 3-6 Number & type of missions detected with DBSCAN

Scenario	DBSCAN Parameters		Total Clusters	Mission Detected	Mission Patterns Detected		
					Covid Samples Delivery	Package Delivery	Fire Surveillance
	eps	minPts					
0	0.6	3	11	9	7	2	0
1	0.6	3	9	7	5	2	0
2	0.6	3	7	5	4	1	0
3	0.4	3	12	7	5	2	0

3.5.3 UTM Air Traffic Flow Characterization

This section presents a characterization of UTM air-traffic flows, with the help of the above mission clusters, as detected by DBSCAN. Three key characterization parameters (KCP) will be introduced, namely: Distance from Centroid (DFC), Distance To Complete Mission (DTCM), and Time To Complete Mission (TTCM). These parameters present the traffic-flow behavior and efficiency of the UTM

airspace in three different scenarios, as explained in the simulation section. DFC is the measure of the horizontal deviation flows, whereas the DTCM and TTCM address the efficiency of UTM air traffic. These model parameters may facilitate the derivation of future configurations of UTM airspace structures. The concept of distance from centroid is inspired by several studies, as referenced below.
One recent article [204] uses the concept of convex polygon and centroid detection for UAV obstacle avoidance. In the pursuit of enhanced path planning for UAV search-and-coverage missions in previously unexplored areas, the research conducted in [205] employs the distance from the centroid of the uncovered sections within the boundary search region. In their study, Lalak and Wierzbicki [206] employ the distance from centroid technique to detect and classify aviation objects using imagery data.

This approach is also employed in the present study, where distances from the mission polygon centroid (DFC) to every point along the trajectory are measured to assess deviations in missions across the entire trajectory. The concept of DFC measurement for shuttle-service missions and trajectory deviations is presented in Figure 3-24.

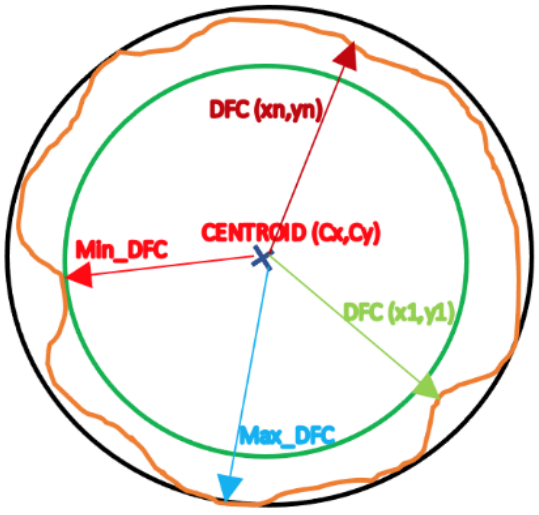

Figure 3-24 The concept of DFC measurement

The centroids (C_x,C_y) are calculated using the area-of-mission polygon. The mission is represented by orange-colored contours. The black and green circles

represent the maximum and minimum distances from centroids, namely, max_DFC and min_DFC, respectively. The DFC $(x_{1,}y_1)$ and DFC $(x_{\mathrm{N},}y_{\mathrm{N}})$ represent the distances from the centroid for trajectory points. The maximum, minimum, mean, and standard deviations were reported in three distinct scenarios to illustrate the impact of dynamic obstacles, weather constraints, and high-priority missions on the observed deviations. These factors indirectly affect efficiency, and they also highlight traffic-flow behaviors in UTM shuttle-service-based missions. Furthermore, a statistical analysis of Distance from Centroid (DFC) was conducted across various mission types, allowing for the correlation and inference of the specific shuttle-service missions that had an adverse impact on traffic flow.

The distance from centroid (DFC) plays a crucial role in understanding UAV mission-path deviations and identifying opportunities for future capacity enhancements, especially for time-scheduled shuttle-service missions, and it plays a key role in dictating UTM traffic flows. This is the case, because shuttle-service missions of the same, or different, UAVs at a depot try to follow the same optimized planned trajectory in different time frames at margins of twenty minutes' delay, as simulated in this study. Nonetheless, the presence of dynamic obstacles, weather conditions and emergency operations prevent this, and deviations are forced that must then be evaluated by the UTM authority. Furthermore, it should be understood that the concept of traffic-flow behavior is slightly different from that of the ATC domain, as most urban-operation UAVs may operate in an airspace zone of 60 km x 60 km or less, without any terminal-maneuvering-area (TMA) requirements. This is especially true for vertical take-off and landing (VTOL) UAVs. Moreover, a number of the UAV missions are closed-loop scheduled missions, which do not evince any merging or leaving trends. This makes it necessary to model the deviation trends using DFC statistics.

The Euclidean distances from the centroid of the closed polygons, representing both the NHS Covid-19 sample delivery and the package-delivery service missions of UAVs, were utilized as a modeling parameter, referred to as Distance

from Centroid (DFC). The centroid is also known as the “center of gravity” or the “center of mass”. This assumes the polygon is closed, so that the last point position $(x_{N,}y_N)$ is the same as the first $(x_{1,}y_1)$, where N is the number of points in the mission trajectory set, or in mathematical-shape terms, the number of vertices in the polygon. The x and y co-ordinates of this centroid (C_x,C_y), for a polygon of this kind, are governed by the following set of equations [204], [207]:

$$C_x = \frac{1}{6A}\sum_{i=0}^{N-1}(x_i + x_{i+1})(x_i y_{i+1} - x_{i+1}y_i) \quad \textbf{(3-7)}$$

$$C_y = \frac{1}{6A}\sum_{i=0}^{N-1}(y_i + y_{i+1})(x_i y_{i+1} - x_{i+1}y_i) \quad \textbf{(3-8)}$$

where x and y are the co-ordinates of our trajectory $T(x,y)$. A is the area of the mission polygon. The polygon is comprised of line segments between N vertices $(x_i,\ y_i)$, $i = 1$ to N. The position of the last vertex $(x_N,\ y_N)$ is assumed to be the same as the first (i.e., the polygon is closed). Subsequently, the area of this closed polygon is calculated using the following equation:

$$A = \frac{1}{2}\sum_{i=0}^{N-1}(x_i y_{i+1} - x_{i+1}y_i) \quad \textbf{(3-9)}$$

In this study, the centroids of the mission polygons were calculated using the built-in function "centroid" in MATLAB. The Distance from Centroid (DFC) is a Euclidean distance between the centroid of the mission polygon and current trajectory point $T(x,y)$, as shown in the equation below:

$$ADFC(C_m, T_j) = \sqrt{(C_{mx} - x_j)^2 + (C_{my} - y_j)^2} \quad \textbf{(3-10)}$$

where C_m is the centroid for any mission (m), and C_{mx} , C_{my} represents the x, y co-ordinates of the centroid. The T_j represents the j^{th} point of the trajectory, with $(x_j,\ x_j)$, co-ordinates. Once the centroid was known, the Euclidean distance (DFC) between the centroid coordinates C_x , C_y and each point along the mission

trajectory was calculated. This was done to analyze the deviation trends, using the statistical measures of mean, max, min and standard deviation (mean_DFC, max_DFC, min_DFC, std_DFC) for each mission detected by DBSCAN. Furthermore, in order to characterize the traffic-flow behavior based on distance to go, the Distance To Complete Mission (DTCM) was calculated ; this was because ideal planned distance might differ from the real planned distance in the presence of the aforementioned constraints. In this study, the UAV speed (Uv) is assumed to be constant (90 km/hr), which means that the "Time to Complete Mission" (TTCM) has a linear relationship with the "Distance to Complete Mission" (DTCM), scaled down by a factor of constant velocity. The mathematical representation of these two parameters is presented below:

$$DTCM = \|T_i - T_{i+1}\|_2 = \sqrt{\sum_{i=0}^{n-1}(x_i - x_{i+1})^2 + (y_i - y_{i+1})^2} \quad \textbf{(3-11)}$$

Where n is the number of trajectory points in the mission

$$TTCM = \frac{DTCM}{Uv} \quad \textbf{(3-12)}$$

The aforementioned parameters were measured and presented for three simulation scenarios, including the ideal scenario-0, which assumed ideal weather conditions and the absence of no-fly zones.

3.5.3.1 UTM Air Traffic Flow Characterization- Scenario- 0

This section includes the results and discussion concerning the KCPs (DFC, DTCM and TTCM) for the ideal case scenario, *with* normal weather conditions, and *without* NFZ restrictions.

a) Distance From Centroid (DFC)

The detected mission polygons with centroids for both NHS Covid-19 sample delivery and commercial package-delivery services are presented in Figure 3-25. Subsequently, the statistical distribution of DFC for all missions was presented

by analyzing the mean, maximum, minimum, and standard deviations of the DFC curves.

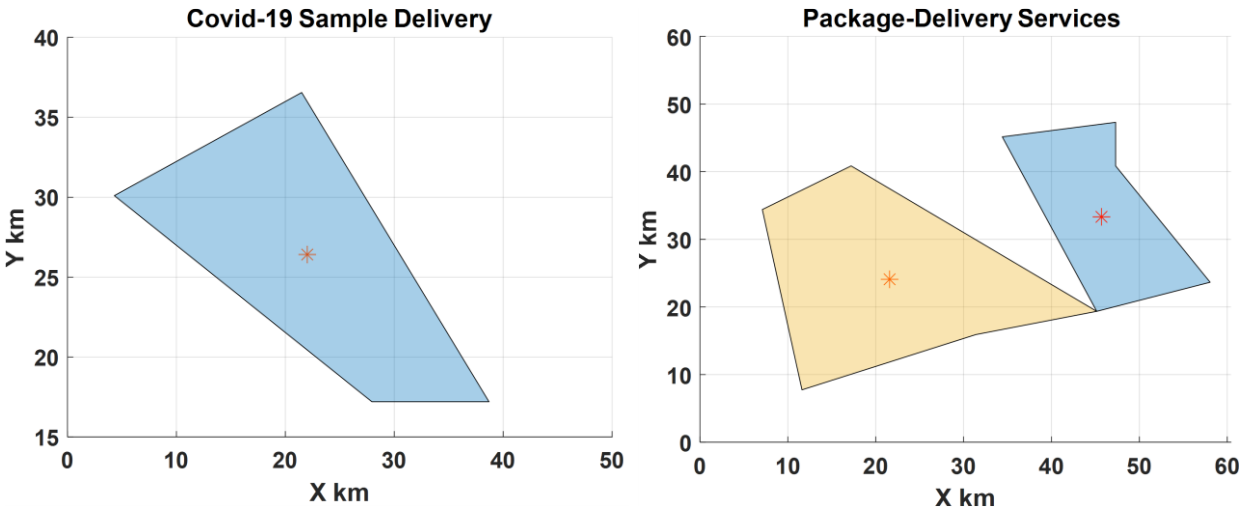

Figure 3-25 Scenario-0: Covid-19 sample delivery (left) and delivery services (right) mission polygons with Centroids (*).

In addition, this analysis evaluates the deviations of the above factors, along with the deviation of centroids for each detected mission for NHS Covid-19 sample delivery (NHS) and the package-delivery services (DS). The results are presented graphically in Figure 3-26.

It is clear from the above analysis that in the ideal case scenario, UTM traffic-flow patterns, as identified by DBSCAN, exhibited zero deviation of DFC values for all the high priority NHS Covid-19 sample-delivery missions, as well as the Package Delivery services. Thus, a clear, sunny day, without any NFZ constraints, exhibits a zero deviation in mean, max, min and standard deviations for DFC. Furthermore, the (x, y) centroids of the described missions retained their constant positions in this scenario. This means that every group of shuttle-service missions followed the same path, without any deviations.

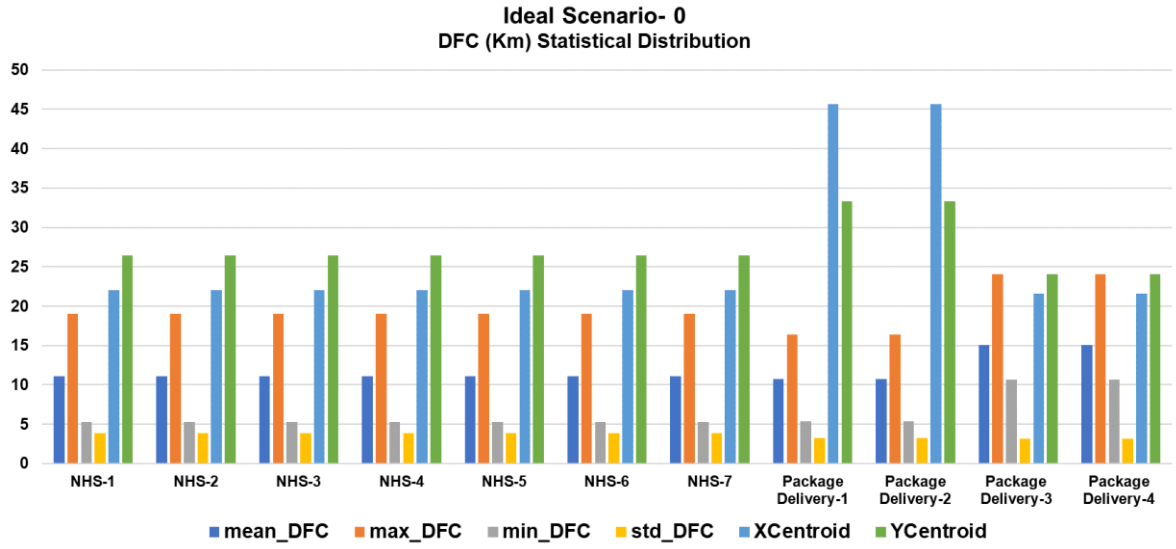

Figure 3-26 Scenario-0: NHS and delivery services (DS) missions DFC Statistical Distribution

b) Distance To Complete Mission (DTCM) and Time to Complete Mission (TTCM)

This section provides the DTCM and TTCM analysis for the ideal case scenario, whereby the DTCM and TTCM metrics are calculated using (3-11) and (3-12). A constant UAV speed of 90 km/hr is assumed in these calculations. Figure 3-27 demonstrates that zero deviation in DTCM and TTCM values was observed for all the high-priority NHS Covid-19 sample-delivery missions, and also for the commercial-delivery services, in the case of moderate weather without any NFZ constraints.

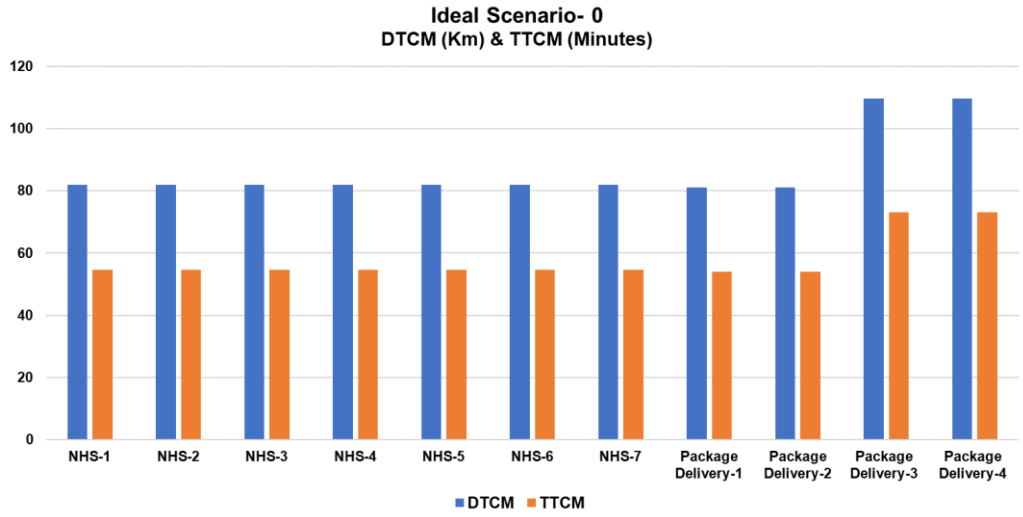

Figure 3-27 DTCM & TTCM distribution for Scenario-0

3.5.3.2 UTM Air Traffic Flow Characterization - Scenario- 1

This section includes the results and discussion regarding the KCP (DFC, DTCM and TTCM) for Scenario-1, whereby the Bedfordshire area was assumed to have an active static NFZ, without any dynamic obstacles or weather constraints.

a) Distance From Centroid (DFC)

The detected mission polygons with centroids for NHS Covid-19 sample delivery and the package-delivery services are presented below, in Figure 3-28. Subsequently, the statistical distribution of DFC for all missions was presented. This analysis also assesses the deviations of the above factors for a group of hourly shuttle-service missions. The NHS-1 to NHS-7 shuttle services belong to one group, whereas DS1 and DS2 belong to another group, and DS3, DS4 constitute the same group. The statistical distribution is presented graphically in Figure 3-29.

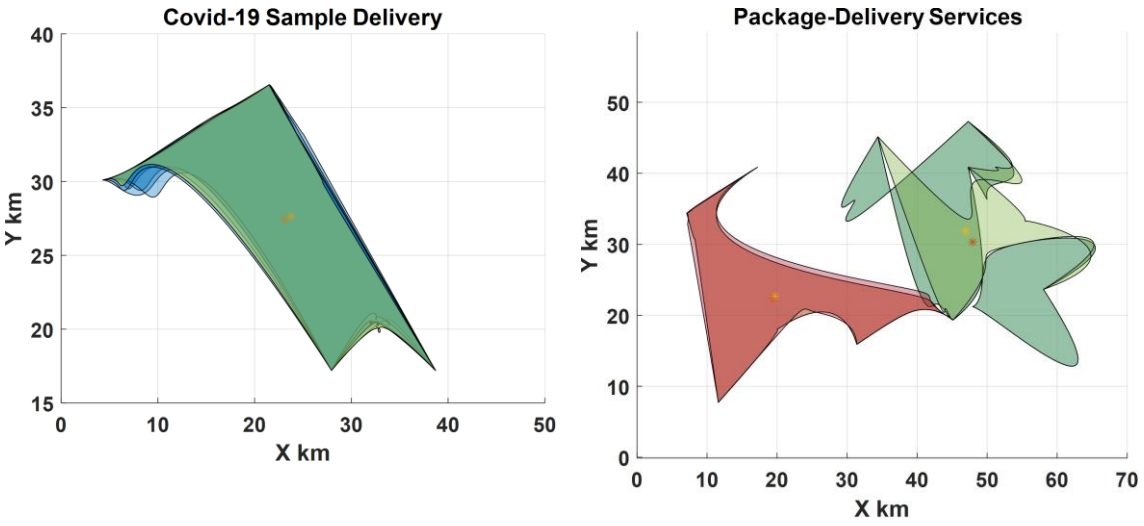

Figure 3-28 Scenario-1: Covid-19 sample delivery (left) and delivery services (right) mission polygons with Centroids (*).

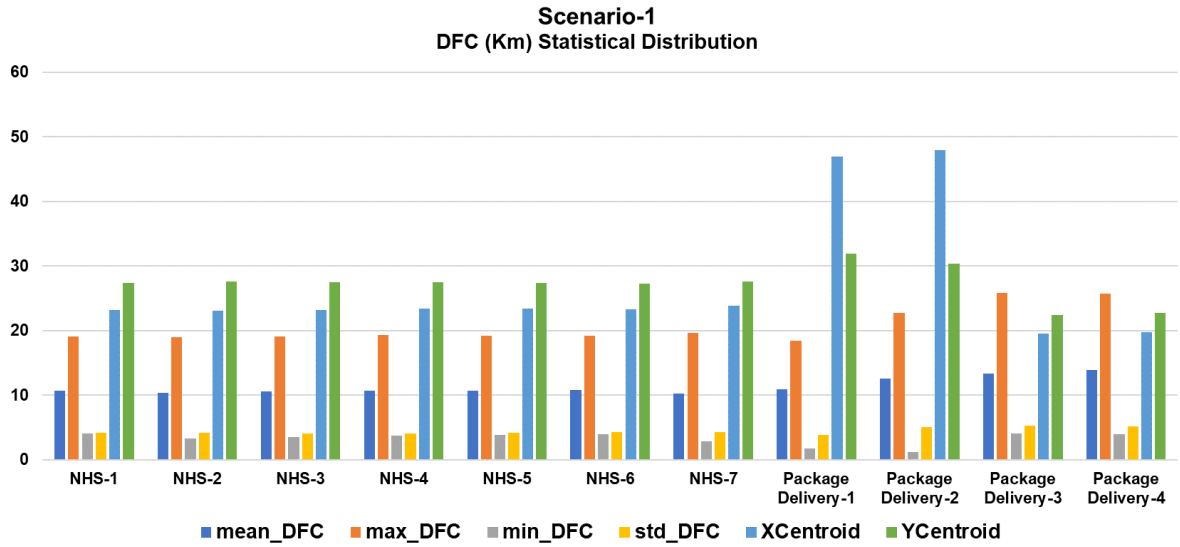

Figure 3-29 Scenario-1: NHS and DS missions DFC Statistical Distribution

It is evident, from the analysis above, that in the case of Scenario-1, traffic-flow patterns show deviations in DFC values for all the high-priority NHS Covid-19 sample-delivery missions, as well as for the commercial-delivery services. The range of deviations in mean, maxima, minima and standard deviations for the DFC curves in this case are as follows. The mean DFC varies between 0.04 km and 1.6 km; the maximum DFC varies between 0.01 km and 4.35 km; the minimum values lie between 0.05 km and 1.14 km; and the standard deviation varies between 0.002 km and 1.16 km. In addition to this spread, a shift is also observed in centroid positions. Specifically, the X centroid varies between 0.03km and 0.96 km, and the Y centroid shifts between 0.02 km and 1.55 km. This means that the NHS and the delivery shuttle-service missions changed their routes in both X,Y planes, and they also increased their spread from the mission polygon centroid by a maximum distance of approximately 4.3 km.

b) Distance To Complete Mission (DTCM) and Time to Complete Mission (TTCM)

This section provides the DTCM and TTCM analysis for the case Scenario-1. The characterization parameters are presented using a stacked bar chart, as shown below in Figure 3-30. Deviations were also observed in the Distance To Complete Mission (DTCM) and Time To Complete Mission (TTCM) for each group of the shuttle-service missions belonging to the NHS emergency services and commercial services. This analysis demonstrated that, in the presence of static

NFZ in the Bedfordshire area, there was some significant deviation of DTCM and TTCM values. For example, the DTCM for the hourly NHS shuttle services varied between 0.3 km and 1.54 km, whereas the DTCM variations for the hourly delivery services shuttle missions ranged between 0.4 km and 20.98 km. The TTCM for the NHS missions varied between 0.24 minutes and 1 minute. The commercial delivery shuttle services exhibited an increase in their mission times from 0.27 minutes to 13.98 minutes.

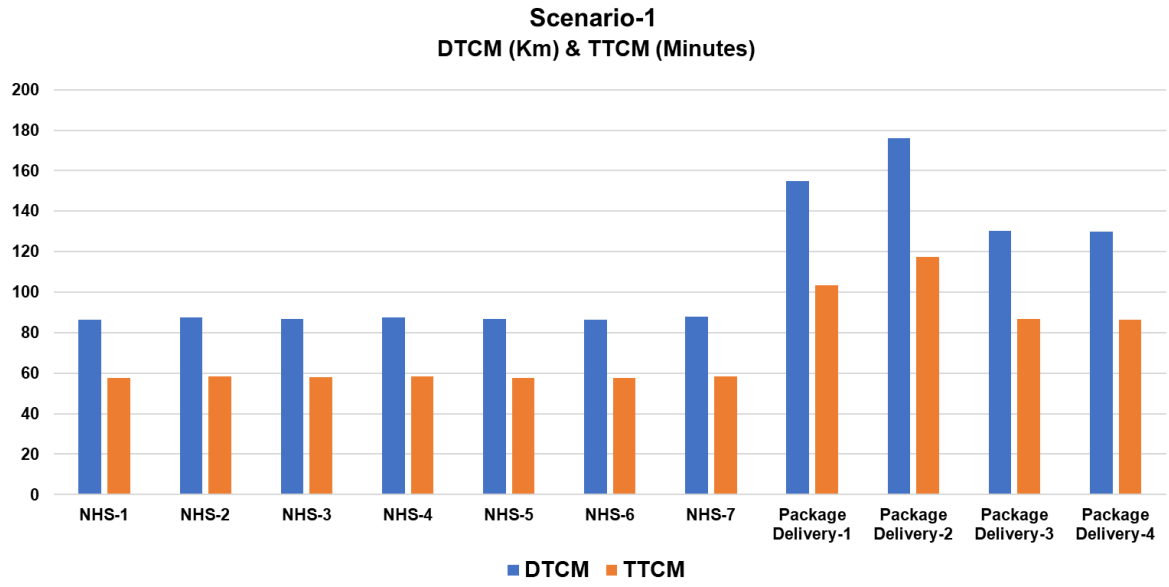

Figure 3-30 DTCM & TTCM distribution for Scenario-1

3.5.3.3 UTM Air Traffic Flow Characterization - Scenario- 2

This section presents the results and discussion regarding the three KCP for Scenario-2. Here, the Bedfordshire area has static airfields and dynamic recreational areas, with extreme weather conditions.

a) Distance from Centroid (DFC)

The detected mission polygons with centroids for all scenario missions are presented below, in Figure 3-31. Similarly, the statistical distribution of DFC for all missions is presented graphically in Figure 3-32.

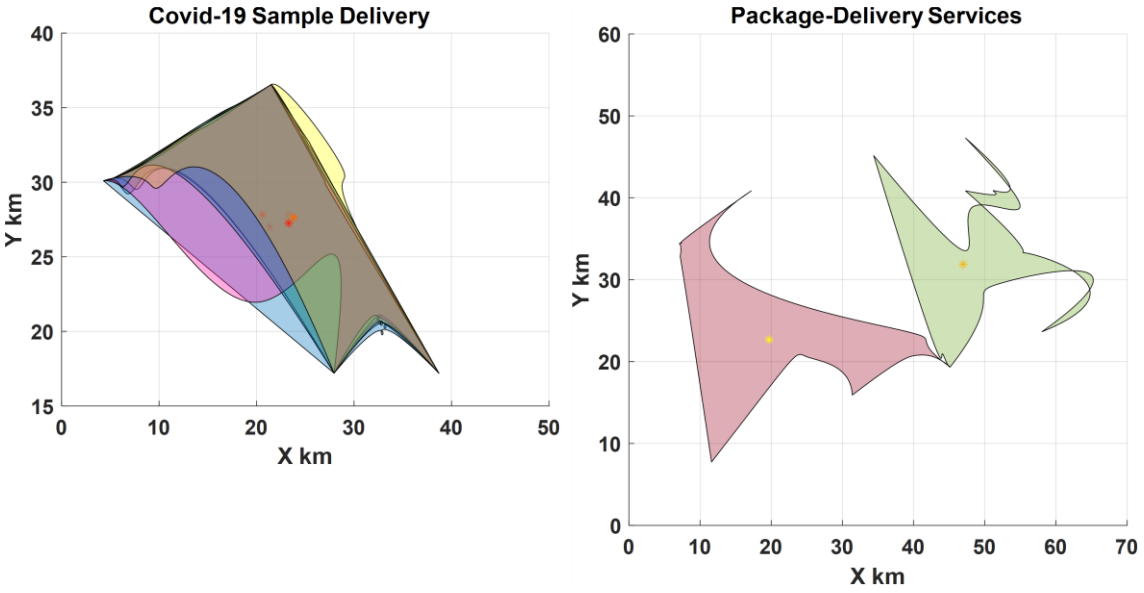

Figure 3-31 Scenario-2: Covid-19 sample delivery (left) and delivery services (right) mission polygons with Centroids (*)

The analysis above demonstrates that, in the case of Scenario-2, traffic-flow patterns evince significant deviations in DFC values for both high priority NHS Covid-19 sample-delivery missions and delivery services. The mean_DFC varies between 0.06 km and 1.19 km; meanwhile, the maximum DFC varies between 0.05 km and 2.059 km, the minimum values lie between 0.128 km and 6.38 km, and the standard deviation varies between 0.005 km and 1.97 km. In addition to this spread, a large shift in centroid positions is also observed; the X centroid varies between 0.108km and 1.88 km, and the Y centroid shifts between 0.2 km and 1.44 km. This means that NHS and the delivery shuttle-service missions not only changed their routes in both X,Y planes, but they also shifted their polygon centroids with reference to the ideal case scenario.

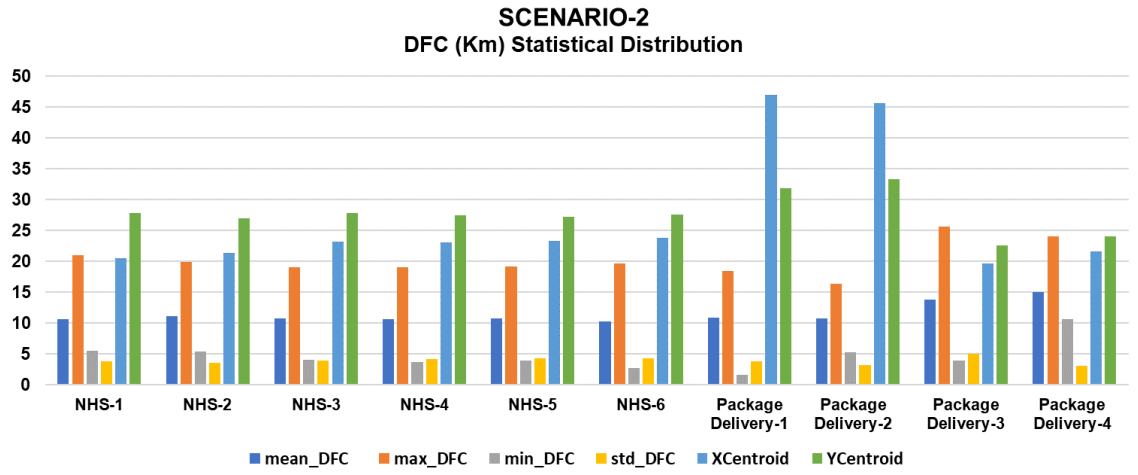

Figure 3-32 Scenario-2: NHS and DS missions DFC Statistical Distribution

b) Distance To Complete Mission (DTCM) and Time to Complete Mission (TTCM)

As indicated in Figure 3-33, deviations can be observed in DTCM and TTCM values for each group of shuttle service, for static recreational areas, dynamic airfields and extreme weather fronts in the Bedfordshire area. The DTCM values for hourly NHS shuttle services varied between 0.82 km and 8.5 km, whereas the DTCM variations for the hourly delivery shuttle-service missions ranged between 20 km to 73 km, when we compared results with ideal case Scenario-0. The TTCM for NHS missions ranged between 0.54 minutes and 5.67 minutes. The package-delivery services missions registered an increase in mission-completion times between 13 minutes and 49 minutes, with reference to ideal case Scenario-0. This clearly shows that the extreme weather fronts, along with the dynamic recreational-area constraints, resulted in elongated mission distances and times, and these factors thus require attention from UTM authorities.

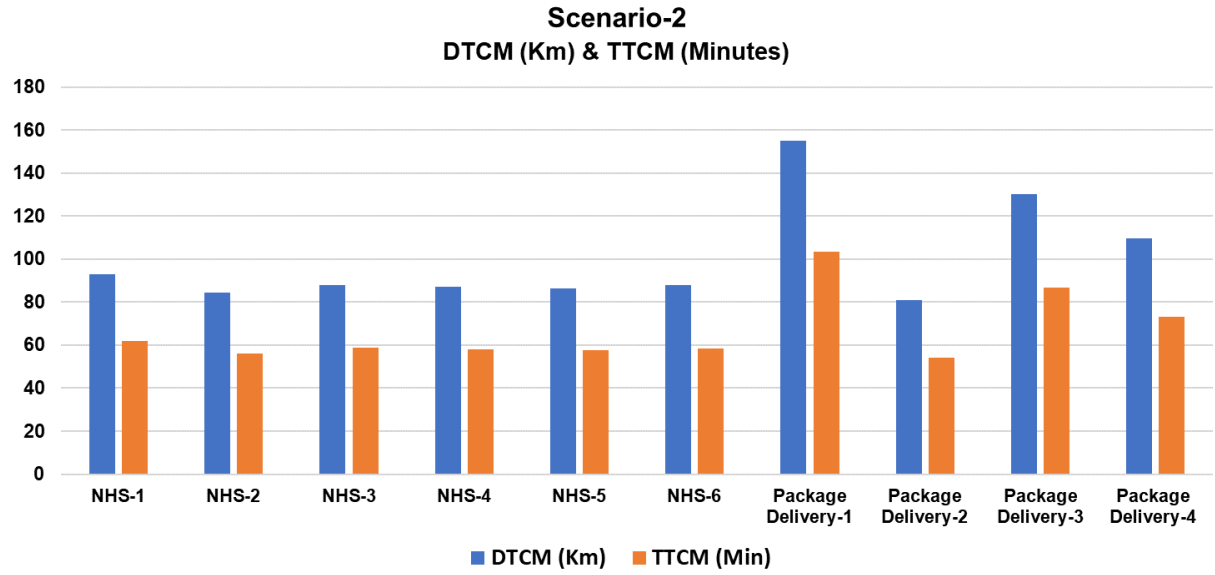

Figure 3-33 DTCM & TTCM distribution for Scenario-2

3.5.3.4 UTM Air Traffic Flow Characterization - Scenario- 3

This section presents the results and discussion concerning the three KCP for Scenario-3, whereby the Bedfordshire area has dynamic airfields and static recreational areas, with “rainy and windy” weather conditions.

1. Distance from Centroid (DFC)

The detected mission polygons with centroids are presented below in Figure 3-34 , and the statistical distribution is presented graphically in Figure 3-35.

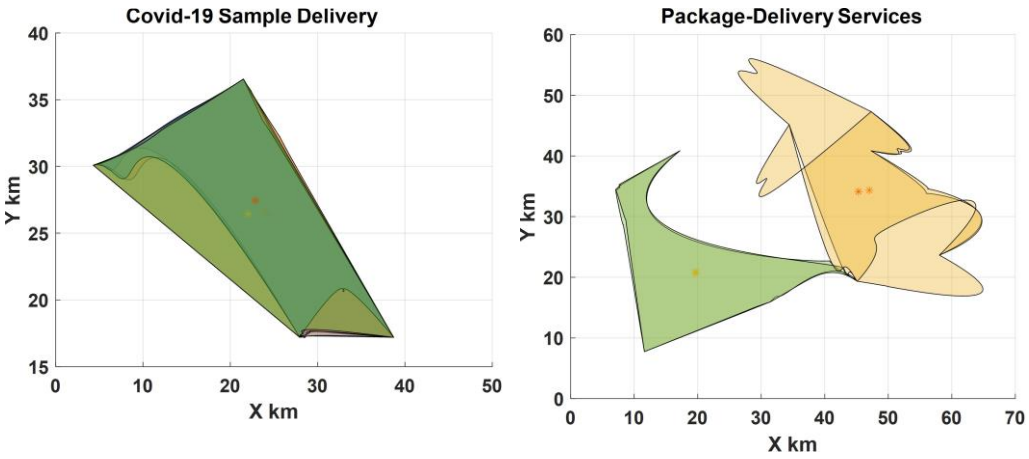

Figure 3-34 Scenario-3: NHS (left) and delivery services (right) mission polygons with Centroids (*)

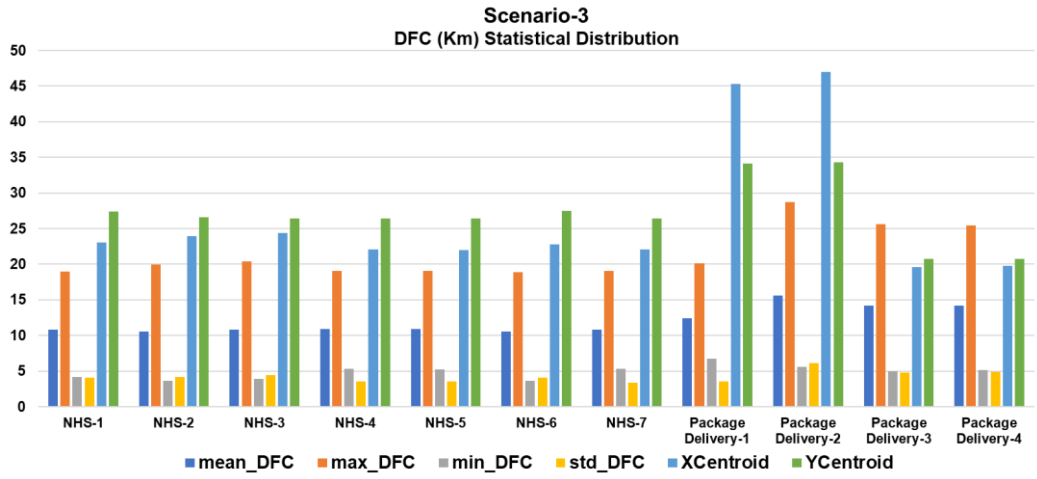

Figure 3-35 Scenario-3: NHS and DS missions DFC Statistical Distribution

It can be seen from the above analysis that, in the case of Scenario-3, traffic-flow patterns evince significant deviations in DFC values. The mean_DFC varies between 0.014 km and 3.20 km, whereas the maximum DFC varies between 0.016 km and 8.6 km, the minimum values range between 0.06 km and 1.62 km, and the standard deviation varies between 0.053 km and 2.593 km. In addition to this spread, a large shift is also observed in centroid positions; the X centroid varies between 0.0.4km and 1.709 km, and the Y centroid shifts between 0.008 km and 1.035 km. This means that NHS and the package-delivery services missions not only change their routes in both X,Y planes; they also shift their centroids.

2. Distance To Complete Mission (DTCM) AND Time To Complete Mission (TTCM)

As Figure 3-36 indicates, more deviations are observed in DTCM and TTCM values for each group of NHS and package-delivery services, due to the constraints imposed by dynamic recreational areas, static airfields and adverse weather conditions, such as rain and wind in the Bedfordshire area.

The DTCM values for hourly NHS shuttle services vary between 0.05 km and 4.99 km. The DTCM variations for hourly delivery shuttle-service missions, meanwhile, range between 0.47 km and 6.91 km. TTCM for NHS missions varies between 0.038 minutes and 3.32 minutes. The delivery services missions evinced an increase in mission-completion times from 0.317 minutes to 4.60 minutes. This shows that rainy and windy conditions, along with NFZ constraints,

also resulted in extended mission distances and times, and these variables will be affected by the efficiency of airspace.

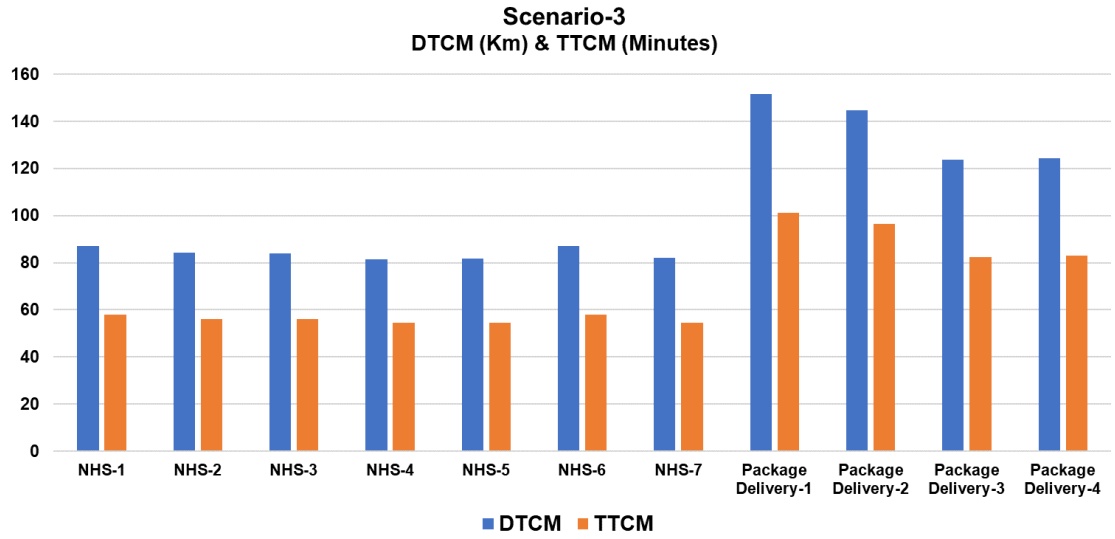

Figure 3-36 DTCM & TTCM distribution for Scenario-3

3.5.4 UTM Airspace Congestion Analysis

This section estimates airspace density in the Bedfordshire area to identify congested regions that may impact UTM airspace safety and availability. The traffic density, which is defined as the amount of aircraft traversing a particular sector, is a conventional metric of air traffic complexity [208]. The density of UAVs' trajectories in an available region has been estimated by analyzing the whole airspace. All scenarios discussed previously have been utilized for the purpose of evaluating the UTM airspace density on the spatial scale. High to low density of UAVs is denoted by yellow to violet colors that mark more populated areas and less populated areas, respectively.

During Severe weather disorders along with emergency operations conducted through UAVs, the UTM operator is responsible for ensuring the safety of flight operations. In the above regard, UTM service providers also judge the capacity of the airspace in order to determine the availability of alternate routes for smooth UTM traffic flow. One of the examples is the diversion of flight movements to off-

peak times or suggesting alternate nearby airports/depots with less congestion or diversion to less congested traffic segments.

The visualize and predict the congestion will help the UTM operator to divert the less priority UAV missions such as recreational UAV flyers to lesser congested areas, in order to regulate the traffic flow and significantly reduce the congestion in the airspace, also offering more availability to upcoming high priority UAV missions. This may help the UTM authority to better manage the airspace safety and availability under real conditions.

The analysis of the data conducted in this study facilitates risk analysis and enhances the planning of trajectories in various aircraft zones taking into account each dynamic parameter that could result in aircraft deviating from their normal trajectory. These findings enable cross-route comparisons of the efficiency of the traffic flow in varying flight situations, in addition to the determination of the factors causing aircraft deviations from their standard routes. Moreover, the airspace density pinpoints some ideal regions to conduct UAV operations that minimizes the impact on conventional ATM. Information pertaining to the flow and performance of traffic sourced from the analysis of trajectory data can lay the foundation for the development of innovative approaches, processes and tools for supporting decisions in the field of air traffic management. For example, this innovation spans from developing an artificial intelligent architecture capable of predicting airspace congestion patterns to classifying safe and dangerous regions using historic UAVs traffic patterns.

In this study, airspace density has been evaluated using the Gaussian Kernel Density Estimator. KDE is frequently employed in the field of computer vision for identifying target objects as in [209]. Additionally, it is utilized for the visualization and analysis of spatial data, with the aim of forecasting event trends [210]. It is also widely applied in areas such as analyzing damage and assessing risk [211].

The kernel density estimation (KDE) based probability density function (PDF) was computed for the bivariate trajectory data (x, y) using the gaussian_KDE() function in the Python. This function automatically determines the estimator bandwidth using Scotts Rule as the default parameter. The computed probability

density function (PDF) was employed to generate congestion contours, which depict the congestion percentages (%) by normalizing the PDF data within the range of 0 to 100. The original PDF lies between 0-1 and represents the per unit increase in kilometers of bivariate data of two trajectory variables x and y. This percentage congestion is plotted as contour maps for the whole span of spatial trajectory data. The results and discussion for KDE and % congestion contours are provided below for each simulation scenario.

A KDE estimation graph for Scenario-1 of UAV missions with static NFZs, no weather constraints and no dynamical obstacles for the first hour (9:00 am to 10:00) is presented in Figure 3-37.

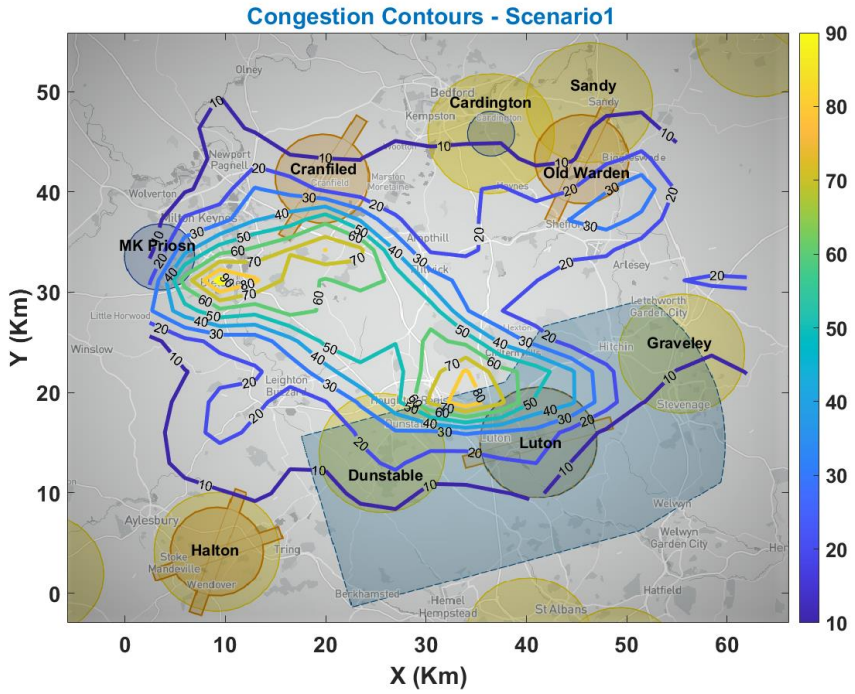

Figure 3-37 Airspace Congestion identification for scenario 1

It has been observed that static areas (NFZs) also exert some impact on the UAV trajectories, resulting in large deviations that cause congestion in different regions of the UTM airspace over Bedfordshire. The regions of Dunstable, Sandy, Cardington and Graveley evince lower regional congestion, as compared with others. It is also observed from percentage congestion contour plots that some

areas near Milton Keynes prison present more than 80% congestion levels. This is a result of local emergency firefighting operations. This observation suggests the need to regulate the traffic flow in these congested UTM traffic-flow zones.

A KDE estimation graph for Scenario-2 of UAV missions with static airfields and dynamic recreational areas with extreme weather conditions for the second hour (10:00 am to 11:00), is presented in Figure 3-38. These dynamics are sporadic, with some regions available and others unavailable throughout the hour.

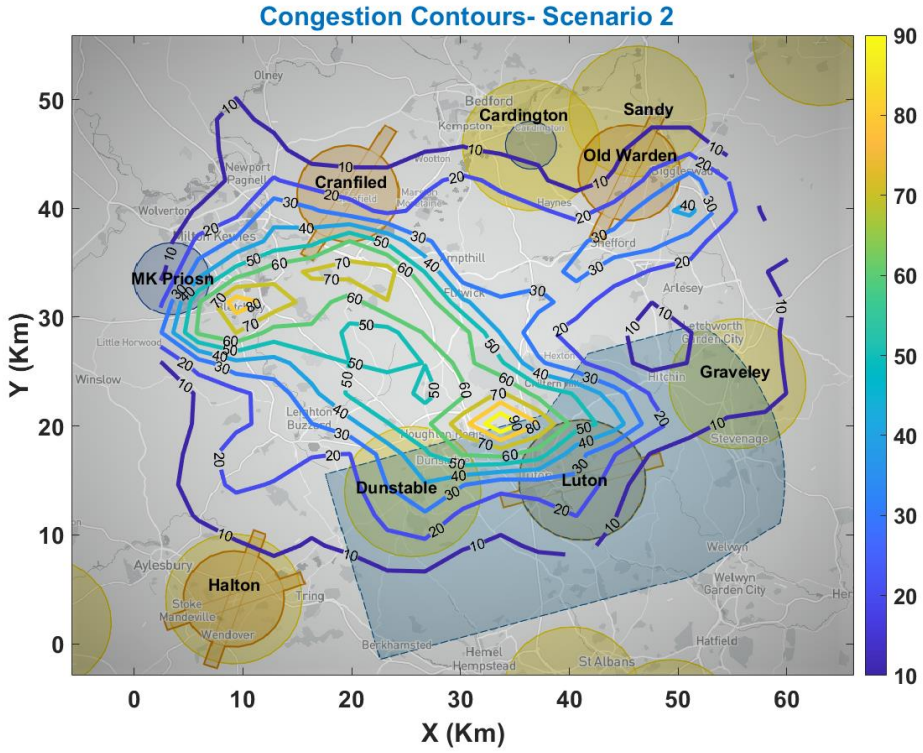

Figure 3-38 Airspace Congestion identification for scenario 2

In this scenario, it has been observed that the density is increased due to a large variation in dynamic and static areas during the hour. Dynamic extreme weather conditions also play an important role in increasing the density of UAV trajectories in some areas as their trajectories deviate to avoid static NFZs and weather fronts. Due to extreme weather constraints, maximum UAVs trajectories confined themselves to a limited area in the whole airspace, as represented by the yellow and green colors in Figure 3-38. It is observed from congestion contour plots that

some areas near Milton Keynes and Luton experience more than 80% congestion levels.

KDE analysis for the third hour of data simulation with dynamic airfields, static recreational areas and different weather conditions (rain and wind) is presented in Figure 3-39.

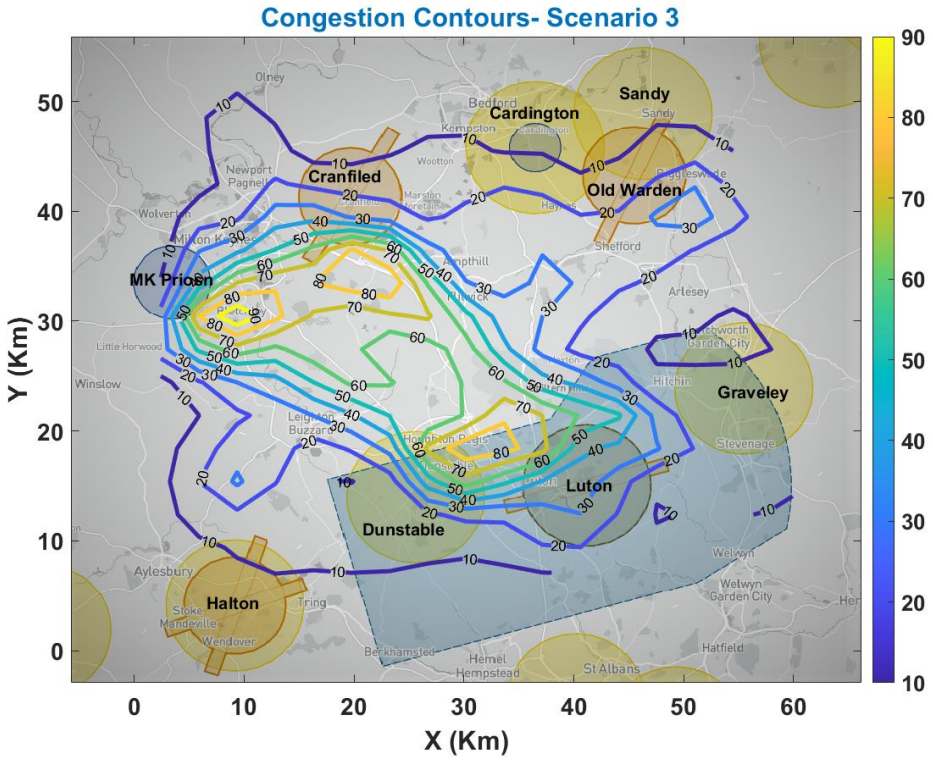

Figure 3-39 Airspace Congestion identification for scenario 3.

It is observed from congestion contour plots that some areas near Milton Keynes, Luton and Dunstable experience more than 70% congestion levels. There is also an observation of about more than 50% congestion levels in some parts of north-west and south-west of Bedfordshire area and thus UTM traffic flow may be regulated in these zones.

It is thus inferred that regions in the north- west and south- east are more prone to hazards and conflicts, as both areas saw maximum UAV-trajectory congestion exceeding 50%, reaching a maximum of 90% congestion levels for some smaller

areas. The reason for the lower congestion of the north-east and south-west regions, conversely, is that they are generally occupied by static and dynamic NFZs with extreme weather constraints; therefore, most UAVs keep their trajectory points away from these regions, avoiding NFZs and extreme weather, leading to more UAV congestions in the remaining airspace regions. This also explains why only the fire-surveillance missions by UAV-6 fail to be detected by the DBSCAN algorithm, as this is considered a noisy pattern and is thus lost in this congested area. The temporal weather effect shows that extreme weather fronts and adverse wind and rain cause some local hotspots, with congestion levels more than 90% as compared to normal weather conditions.

3.6 Conclusion and Future Work

This work has proposed and implemented a framework for UTM traffic flow's spatial and spatiotemporal patterns identification and characterization using DBSCAN clustering algorithm. Moreover, this work has analyzed the airspace density using the Gaussian kernel density estimator (KDE) for different UAV missions under variable airspace structures and environmental factors.

UAV missions considered in this study as an application use case are essential delivery services for COVID-19 samples, package delivery services and emergency fire surveillance tasks. This work also considers the influence of random flights of UAV hobbyists or recreational users during the execution of above essential missions. The effect of airspace structure configurations like static NFZs, dynamic airfields, recreational areas and environmental factors such as weather conditions including wind and extreme weather have also been incorporated and studied in this work.

It has been observed that the unsupervised learning algorithm DBSCAN can detect and identify 82% of UAV missions under ideal weather conditions and in absence of flying restrictions. The valid missions' detection capability of DBSCAN is reduced by 4% under NFZ restrictions and by 10% in the ideal scenario with static airfields, dynamic recreational area restrictions and extreme weather fronts. This performance is further deteriorated by almost 23% with static recreational areas, dynamic airfields and bad weather conditions including wind and rain. The

noisy clusters incorporate almost all single-leg trajectories belonging to recreational users.

It has been observed that types of mission such as COVID-19 sample delivery and package delivery that run as shuttle services were identified in large numbers as compared with the one-time fire surveillance mission because this mission is in the region of maximum interference or congestion, and there is a greater concentration of random trajectories of hobbyists' UAVs as presented in the KDE-based density graphs. This phenomenon resulted in the declaration of such a square polygon as an outlier cluster pattern.

It is also evident from the detected mission-cluster patterns that the ideal trapezoid trajectory polygon shapes, as expected in ideal case Scenario-0 (Figure 3-9 and Figure 3-10), are changed and deformed (Figure 3-14 , Figure 3-18 and Figure 3-22), showing the presence of NFZs, obstacles and weather constraints. The packages-delivery-mission patterns seem most affected; these show far more diversions and present "flower-petal" shapes, as compared with the ideal Scenario-0. The reason for this is that this mission is obliged to provide emergency services, when NFZs and dynamic obstacles are present along with the worst weather conditions.

It can also be concluded from the UTM mission characterization that severe weather conditions and dynamic NFZ are the key sources of UTM-mission unpredictability in the Bedfordshire area. It is observed that UTM missions disperse more in the airspace, with a maximum spread of about 8.6 km. Furthermore, there appears to be a shift in the mission centroids by a maximum margin of 1.88 km in horizontal directions, and of 1.55km in a southerly direction. Indeed, there seems to be increased shifting of missions in the horizontal directions. The commercial package-delivery services are subject to stronger adverse effects, in comparison with the emergency NHS Covid-19 sample-delivery services.

Moreover, the mission ranges increased drastically, by 20 km to 79 km in the worst cases, as compared with ideal weather conditions without any NFZ restrictions. Similarly, the mission-completion times also increased by an order of 14 to 49 minutes in the worst-case scenarios.

For a better congestion analysis using KDE, the percentage congestion contours are evaluated to provide better suggestions for UTM operator as visual data analytics to regulate better traffic flow based on congestion figures. The hot zones with more than 80% congestion levels in Bedfordshire area are pinpointed in this study using the above approach.

It is inferred from the KDE analysis that the north-west and south-east regions are more congested, as compared with other regions of Bedfordshire airspace. The DBSCAN clustering algorithm thus detects UAV trajectory patterns where there is less congestion as compared with densely packed trajectory zones due to random single-leg missions of UAV-flying hobbyists. It is hence concluded that congested areas depict conflicts and concerns related to the safety of the airspace.

It is thus inferred that clustering algorithms can be good candidates in classifying and detecting up to 82% of the UAV mission trajectories, and these statistics reduce to about 58% in the worst-case scenarios. The performance of detection is deteriorated due to the presence of noisy trajectory patterns belonging to random UAV flights of hobbyists, dynamic airspace structure and weather constraints. Moreover, Gaussian kernel density-based estimators may help in evaluating the congestion in the airspace.

The forthcoming research entails the development of an AI-based architecture to predict airspace congestion patterns and classify regions as safe or dangerous. This will be accomplished by leveraging historical traffic patterns of unmanned aerial vehicles (UAVs). Additionally, the application of this research is being considered for identifying UTM airspace patterns and measuring congestion in other regions of the United Kingdom that exhibit diverse airspace configurations and anticipate higher UAV traffic volume. Furthermore, future work aims to design more realistic and intricate operational scenarios. The insights gained from these studies, pertaining to traffic flows and performance analysis, can serve as a foundation for devising novel approaches, practices, and decision support tools to enhance air traffic management.

4 DEEP-LEARNING ARCHITECTURE FOR UAV TRAFFIC DENSITY PREDICTION

4.1 Introduction

In previous studies, the concepts of unmanned traffic management (UTM) [20] and urban air mobility (UAM) [212] have been proposed to carry out safe and efficient aerial vehicle operations. The same studies highlight the fact that extant air traffic systems have seen a paradigm shift in consequence of new UAM and UTM systems. Not least, the latter have presented important challenges for aircraft efficiency and safety. These challenges stem from a range of factors, notably: increasingly dense vehicular operations; difficult lower-airspace meteorological conditions; the characteristics of vehicles themselves; and the nature of urban terrain.

Air traffic flow management (ATFM) is one component of ATM service structure [213], and as such, it must support safe and efficient air traffic operations via the key airspace objective of demand and capacity balancing (DCB) [214]. When based on Artificial Intelligence (AI) algorithms, airspace-management models and ATFM have evinced formidable efficacy in mitigating both delays and congestion [215]–[217]. Over the last few years, the research community has focused heavily on predicting air traffic flows, and this has yielded remarkable outcomes. Conversely, few studies have examined the prediction of air traffic flow for UTM systems. Problems surrounding UAS traffic flow management (UTFM) are ultimately extensions of the more traditional challenges pertaining to standard ATFM. The main concern of the latter is the anticipation of traffic volumes and the prediction of airport resources and airspace usage in relation to current capacity. This, in turn, permits the proactive management of congestion. Either prior to take-off (on the ground) or during flight (via speed changes or airborne holds), delays may then be assigned to aircraft as required, so that resource capacity is not exceeded. As compared with those used for traditional commercial flight, however, new algorithms are needed to accommodate the special characteristics of UTM. Here, elements of difficulty include greater operational density, higher operational numbers, performance discrepancies between vehicles and

operators, and lower operational attitudes [13]. Extant air traffic control (ATC) systems will struggle to accommodate such challenges, and major systemic modifications are thus needed. In particular, if drone flights are not managed by a system incorporating a high degree of automation, both the privacy and the safety of local citizens will be compromised [14].

As the central factor within ATFM, ATFP has been widely researched While research on the prediction of air traffic flow for UTM systems is limited. Most researchers have focused on predicting future traffic densities based on historical data [218]. Moreover, they assume a static environment, with fixed start and destination points for vehicles, as well as fixed airspace constraints regarding no-fly zones (NFZ). A static environment is not a sufficient paradigm to test UTM systems, due to the dynamicity of the UAV operational environment. In the development of an ATFP model, there are two factors that influence air traffic flow in a given section of airspace. These are spatial and temporal correlation [219]. Spatial correlation is the evolution of the motion of an aircraft in neighboring regions (i.e., aircraft in nearby regions may fly into the airspace in question, depending on their future heading). Conversely, temporal correlation addresses the situation of former instants in the current area (i.e., aircraft may be present in the “current airspace”, or they may move to adjacent airspaces).

The frequency of aircraft passing through a given region, over a certain period of time, is known as air traffic density, or congestion [220]. The increase in complexity can negatively impact controller’s decision-making abilities, resulting in increased errors [221]. Thus, so-called “hot spots” of air traffic congestion necessitate scrutiny from controllers to determine whether conflict avoidance or modification is required for any planned trajectories. The literature provides several metrics, developed and implemented for the purpose of measuring air traffic complexity. These include convergence and geometric metrics [222], a clusters metric [223], the Grassmannian metric [224], and finally, the König metric [225]. Whilst these metrics have proven effective in basic scenarios, they are nonetheless unsuitable for complex and/or large-scale applications. The methodologies also fail to take proper account of spatiotemporal data [226]. The present study sought to overcome these difficulties via the adoption of an air

traffic complexity metric centered upon the Linear Dynamical System (LDS). Via analyses of traffic structures and airspace geometries, numerous studies have analyzed traffic structures and airspace geometrics to demonstrate the efficacy of LDS as an intrinsic complexity measurement [222], [227], [228]. Moreover, LDS is appropriate for use as an estimation metric for local disorder and the interaction of trajectory sets within the UTM system. Its suitability derives from its efficiency and relative simplicity, and the fact that it evinces mathematically predictable behavior.

Recently, the control and management of transportation has become more data-driven, largely as a result of proliferating traffic-sensor technologies. Many extant systems and models for traffic flow prediction, however, use shallow traffic models that perform poorly [229]. Conversely, due to recent advances in machine-learning (ML) techniques, air traffic-congestion prediction has been directed towards the utilization of such technology. Deep learning (DL), a subcategory of ML, has attracted huge interest in both academic and industrial arenas [230] due to the successful implementation of various applications, such as natural-language processing, motion modelling, dimensionality reduction, classification tasks, and object detection [231]. Moreover, multi-layer/deep-architecture DL evinces superior performance in the extraction of inherent features in the training dataset; it discovers multiple structures within the data.

In Chapter 3, a Gaussian kernel density estimator (KDE) was utilized to analyze airspace congestion, aiming to identify areas of high activity and regions where interference may occur. However, it is acknowledged that more comprehensive congestion metrics are necessary to accurately represent the level of UTM airspace congestion. Considering this, Chapter 4 proposes the use of intrinsic complexity metrics as a measure of congestion, which takes into account the position and speed of neighboring UAVs. These complexity metrics aid in predicting future trajectories that are likely to form in the vicinity of the UAV under analysis, thereby providing a more accurate assessment of UTM airspace congestion. Furthermore, a more intricate case study model is employed to

generate data for this prediction model, building upon the idealized transport model discussed in Chapter 2. The main contributions of this chapter are:

1. The work adapted an intrinsic complexity metric based on the linear dynamical system model to assess congestion in UTM operations. In this context, the study developed an optimal air traffic assignment model that computes and measures air traffic flow complexity in the neighborhood of a UAV at a given time. The proposed strategy explicitly considers operational differences between ATM and UTM systems, such as dynamic flow structures, airspace density, separation requirements, and standards.
2. To validate the proposed model, three different practical drone-delivery scenarios were conducted in the simulation-scenario environment. The missions spanned a wide range of applications with variable numbers of UAVs. Furthermore, the effects of airspace-structural configurations, such as static NFZs, airfields with variable availability for drone flights, recreational areas, emergency UTM operations, and environmental constraints such as weather conditions were studied.
3. The study found that the existing literature in this field covers either trajectory prediction or conflict detection and resolution, with limited research on the prediction of air traffic flow for UTM systems. To address this gap and improve the safety and efficiency of UAVs operated in urban areas, the present study proposed a learning-based model to predict air traffic congestion over a period of three minutes. The proposed model was adapted to make it suitable in terms of the look-ahead time horizon of the UTM applications (such as drone delivery services, emergency operations, and inspection-related UAV tasks). With the information supplied by such a congestion prediction system, it will also be possible to plan a safe flight trajectory in advance.
4. A critical aspect of UTM operations centers on complexity assessment, and the computational effort that the latter entails. This is particularly true for on-board applications. A key goal of the present research, therefore, is to render the traffic prediction model significantly ‘smaller’ without

surrendering any notable degree of predictive accuracy. In terms of methodology, the study took deep learning-based predictive models from other fields and reconfigured them. These models included ATFP for aircraft and road traffic prediction. Finally, a DL model comprising both LSTM and 1-D convolutional neural networks (1D-CNNs) is recommended. Indeed, the capacity of the proposed model to extract spatiotemporal components from UAV flight data, and to do so within an acceptable computing timeframe, is confirmed by the experimental findings.

The structure of this chapter is organized as follows: Section 4.2 presents related work and traditional methods for traffic flow prediction. Section 4.3 discusses the proposed methodology of this work. Section 4.4 discusses the proposed prediction model results. Lastly, Section 4.5 provides a conclusion of this chapter and some direction for future work.

4.2 Related Work

ATFP is a vital element within contemporary air traffic research and seeks to predict traffic characteristics at (1) a given point in the future, and (2) for specific airspace. It does this in relation to previous operational data and the traffic context in real time [219]. In studies of ground transportation systems, traffic flow prediction has greatly evolved in conjunction with advancements in information technology. In studies focusing on air traffic, however, the newest models have not been applied due to concerns around safety [232]. A vast range of modelling techniques have been innovated and dedicated to the development of ATFP models and can be essentially classified into four cohorts: flight plan-based algorithms, methods based on statistics, and traditional methods of ML and DL algorithms, respectively.

As a two-fold ATFP method, flight plan-based algorithms involve the deployment of a four-dimensional trajectory forecast to assess a flight in relation to its flight plan via individual waypoints. These algorithms are then used for air traffic flow prediction within the airspaces involved [233]. A shortcoming of this model,

however, is that it does not incorporate information relating to the traffic context in real time. Moreover, its efficacy depends on the four-dimensional trajectory projections being sufficiently precise. One proposal is for predictions to be improved by the addition of real-time, flight-position assessments [234]. Nonetheless, since flight-trajectory predictions are incapable of incorporating rapid, real-time traffic changes, the predictive outcomes still diverge markedly from actual values.

Statistics-based methods are composed of widely used linear stochastic models such as the moving average (MA), autoregressive, autoregressive integrated moving average (ARIMA), seasonal ARIMA, autoregressive fractionally integrated moving average, autoregressive conditional heteroscedasticity (ARCH), and the generalized ARCH [235], [236]. These classic time series models are able to predict the temporal dependencies in time series data; however, they exhibit poor prediction of the spatial influence in urban flow prediction problems [237]. Nonetheless, ARIMA is a more useful and straightforward strategy than alternative models because the only information required is prior data. The Hidden Markov Model (HMM) is very useful in traffic engineering map-matching, especially when using probe vehicle data. Sun et al. [238] used the HMM to map the trajectory of GPS points observed in nearby roads. These candidate points were chosen as HMM hidden states and were more likely to be observed because they were closer to the observation point. To avoid misleading results caused by abrupt traffic situations, the transition probabilities of two adjacent candidates were also considered. The HMM demonstrates precision in selecting a traffic pattern or a traffic point. It has the advantage of being able to deal with data that contain outliers. Points with short sampling intervals appear to be well matched, whereas long intervals and higher similar probe data reduce model accuracy.

Shallow machine learning (SML) uses basic and traditional algorithms, which contain few hidden layers, to generate generalized predictive models. SML algorithms are unable to perform the features extraction from the input, thus expert manual definition of features is needed. Well-known SML algorithms incorporate the artificial neural network (ANN), support vector machine (SVM),

linear regression, second- and third-degree polynomial regression, and k-nearest neighbor (KNN), among others. The aim of developing ANN is to solve complex nonlinear problems by simulating the function of the human brain and the nervous system. ANN can accurately extract fine information and deep knowledge from datasets by establishing empirical relationships between input and output variables. Thus, ANN has been commonly employed in the development of numerous transportation models. Karlaftis and Vlahogianni [239] provide an overview of traditional ANN approaches in transportation research. Due to unique features, such as easy integration and efficient predicting ability, ANN has been widely used in research regarding traffic congestion prediction [230], [240], [241]. ANN is a useful and flexible ML model configuration with an ability to adapt according to input data. It does, however, require larger datasets, which results in high complexity [230]. Due to the complex management of airspace, the shallow NN-based model is insufficient for the prediction of air traffic flow and unable to extract high-level transition air traffic patterns. A predictive autoregression model was developed, with a view to improving the accuracy of air traffic predictions. This model comprised a merger of SVM with both a robust autoregression model and a polynomial model—the goal being to produce a coalesced predictive paradigm [242]. For testing purposes, genuine air traffic data were harvested from the Beijing ATC area. Compared with the SVM only, the combination model generated an improvement in ATFP precision of almost 3%.

Recently, Deep learning (DL) algorithms have gained more recognition in traffic spatiotemporal tasks due to the improvement of computing power and big data analytics. DL algorithms comprising of multiple hidden layers to process complex nonlinear problems enable extraction of features from the input data without prior knowledge [243]. Unlike SML, DL algorithms can perform feature extraction and involve in-model training. Therefore, DL has become increasingly prominent in studies predicting traffic flow [244]. The most popular DL-based methods include the CNN and the recurrent neural network (RNN), which are used in computer vision [245] and sequence learning tasks [246], respectively. The main feature of RNN is using an internal memory to produce a new output, together with current

input and stored output from the previously processed input [247]. Therefore, RNN considers each input as dependent on any other, enabling use of input data that are temporally related (i.e., time series). Accordingly, RNN has become one of the most suitable modeling techniques for ATFPs. Its performance, however, is poor when capturing long-term dependencies and vanishing or exploding gradients during the training phase. When processing long time series, this drawback will prevent the network from converging, which has been termed as the vanishing gradient issue [248]. To overcome this issue, many RNN layer configurations with long-term memory (i.e., LSTM) have been investigated [249]. Unlike RNN, the complex LSTM structure allows interactions between current input and all previous inputs [250]. Thus, LSTM is considered to be a potent approach for time series prediction allowing traffic flow forecasting [251]. CNN is a subcategory of DL and widely applied in image processing applications of traffic flow prediction [252]. For instance, CNN has a significant role in analyzing visual images in traffic prediction by converting traffic flow data into a 2D matrix [253].

In recent years, graph neural networks (GNNs) have emerged as the forefront of deep learning research, demonstrating superior performance in various applications [254]. GNNs are appropriate for traffic forecasting problems due to their capacity to capture spatial dependency, which is represented by non-Euclidean graph structures. The researchers of [255] examined modern and advanced graph neural networks that are used in traffic forecasting. The studies evaluated in this survey have been organized based on the type of traffic graph and adjacency matrices employed. Moreover, the airspace structure and flow pathways are both taken into account in the Temporal Attention Aware Dual-Graph Convolution Network (TAaDGCN) proposed in a recent study [256] to predict air traffic flows. The proposed approach uses real-life flight data and can be used to improve prediction performance and render it superior to existing state-of-the-art comparison methods (especially approaches that ignore the importance of the sector spatial structure). In a similar vein, [257] recommends using the graph concept to characterize airports as nodes with time series attributes and perform data mining on graph-structured data. To be more precise, a temporal graph dataset is created by pre-processing airline on-time

performance (AOTP) data. The mobility level at each airport can subsequently be predicted using a spatial-temporal graph neural networks model.

An aggregated ML model, which would address temporal and spatial dependencies in adjacent areas, flight levels, and previous traffic situations, and which (thus) would accommodate the overall air traffic flow situation, was suggested by Lin et al. [219]. Alongside the use of traffic flow-matrix (TFM) data, the integration of CNNs and RNNs (ConvLSTM) for traffic flow prediction allowed a novel, combined ML model to be trained. Compared with earlier approaches, these study results evinced superior performance. This was because the model enhanced ATM operational efficiency by predicting flow distribution at various flight levels. Moreover, an ATFP complexity metric was employed by Shi-garrier et al. [258]. This approach comprised a novel form of innovative encoder-decoder LSTM neural network, while being independent of any traffic control system. In predictions of airspace-complexity values 40 min in advance, the results showed a Mean Absolute Error (MAE) for the proposed model of 0.08. Image-based trajectory data (as inputs to a CNN and LSTM cascaded deep neural network), meanwhile, were deployed by Zhao et al. [218], who consequently predicted UAV instantaneous density. This entailed a segmentation approach, with a concomitant reliance on historical data. This model, however, did not address the kind of practical or realistic mission accommodated in the present study, even though it generated a one-hour continuous prediction time horizon (with good correlation ratings), despite evaluating the proposed network via a correlation metric. Moreover, the Zhao study failed to consider the impact of UAV prioritization, or the influence of airfields, recreational areas, or other dynamical airspace structural limitations. (The need to establish a priority list for various missions remains constant.) Finally, this study also failed to consider the impact of meteorological factors, such as rain, adverse wind conditions, or general ‘extreme weather’.

In summary, the aforementioned model-driven methods can accurately simulate future traffic flow based on the special statistical features of raw data. Although the interpretation of these models is simple, parameter estimation and defining

assumptions are complex tasks. Therefore, few model-driven methods can be applied to stochastic traffic flows or used for generating highly accurate predictions. In addition, whilst combination models show higher accuracy than model-driven methods, they are not widely applied in traffic flow prediction due to their complexity and poor performance in real-time applications. Furthermore, SML failed to consider the full spatiotemporal dependencies of the traffic situation due to the higher complexity of big data and restricted deep mining [237]. Researchers have responded positively, by contrast, to ANNs, since the latter evince both good predictive performance and mature theories. Researchers involved in traffic flow forecasting have also demonstrated increasing interest in LSTM, a superior time series-analytical DL network [244], [259], [260].

4.3 Methodology

Given the lack of UTM historical data, the present study developed and implemented a scenario-simulation framework that takes into account airspace structure, emergency UTM operation, and environmental factors (such as weather conditions). In this study, the simulation scenarios, including weather factors, airspace structures, conflict detection and resolution, formation of UAV states, and calculation of the complexity metric, were performed using MATLAB. On the other hand, the convolutional LSTM neural network was implemented using the Keras framework.

In the following section, the details of the proposed model will be presented, including data generation, the formulation of UAV states, the complexity metric computing, and the architecture of the prediction model.

4.3.1 Data Generation and Simulation Scenarios

In order to assess and verify the method suggested by the present study, a simulation was conducted in the airspace over Bedfordshire, UK. The simulation identified areas where flight is potentially restricted, including four recreational zones (Cardington, Dunstable, Graveley, and Sandy—Yellow), four airfields (Cranfield, Halton, Luton, and Old Warren—Orange), and Milton Keynes Prison (Blue). Validation of the proposed model was undertaken via a drone-delivery system: the latter was employed for package delivery, emergency fire

surveillance operations, an inspection of the railway infrastructure with the use of UAVs, and the essential dispatch of COVID-19 test samples, with differing priority levels assigned to the operations in question.

The first assumption was (a) that a fleet of three UAVs (designated UAV1, UAV2, and UAV3) would be maintained by the Luton (Bedfordshire) National Health Service (NHS) hospital, and (b) that these would be used to deliver sample tests to a range of clinics within a defined distribution network. This UAV shuttle service would be organized on an hourly basis. The second assumption was that mail packages would be delivered to multiple post offices from the Luton central delivery office. Two vehicles would be deployed for this, namely, UAV4 and UAV5. The first would cover Cardington, Graveley, Old Warren, and Sandy, while the second would be allocated to cover Cranfield, Dunstable, Halton, and Milton Keynes Prison. The use of two UAVs in this manner was anticipated to improve customer satisfaction and minimize delivery times.

Camera-based surveillance operations played a pivotal role in the rescue operations pertaining to firefighting. The images and videos obtained from the payload cameras significantly influenced the precise nature of the firefighting response. The scenario under consideration focused on a singular UAV operation. Milton Keynes Prison Fire Station served as the designated base station for the pertinent missions, utilizing UAV6's capabilities. Additionally, an inspection mission was executed to assess the condition of railway tracks in the Bedfordshire area. The primary objective involved examining the technical state of the railway infrastructure spanning from Milton Keynes, passing through Bletchley, and concluding at Bedford Central Railway Station. Two UAVs, namely, UAV7 and UAV8, were allocated for this inspection task. The former covered the track from Bedford rail station, via intermediate waypoints, to a railway station near Ridgmont, then back again to Bedford rail station for the next hour of inspection. Conversely, UAV8 was deployed to cover the track from Milton Keynes (MK) central rail station to Bletchley and back to MK rail station. Inspection took place on an hourly basis. In the scenarios addressed, the consideration was given to the potential employment of UAVs by hobbyists in parallel with the outlined missions. These trajectories were assumed to consist of

a single leg, with starting and finishing points determined randomly. These trajectories were assumed to have one leg, with starting and finishing points determined at random. The missions themselves, and the simulation environment, are illustrated in Figure 4-1, while the plan and schedule for each mission are presented in Table 4-1.

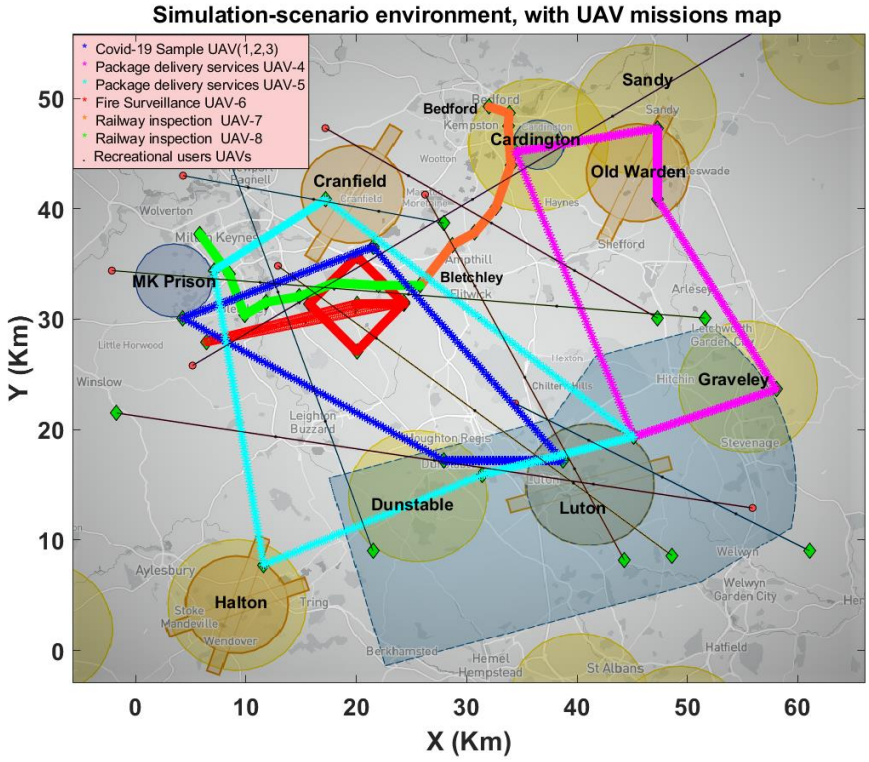

Figure 4-1 Simulation-scenario environment, with UAV missions map

Table 4-1 The description and schedule for UAVs missions

Mission	Covid-19 samples	Package delivery	Emergency operation	Railway inspection
UAV fleet	UAV1, UAV2 and UAV3	UAV4 and UAV5	UAV 6	UAV7 and UAV8
Priority	2	3	1	4
Route	Luton Cranfield Milton Keynes Dunstable	UAV4: Cardington Graveley Old Warren Sandy UAV5 Cranfield Dunstable Halton Milton Keynes	UAV6 available at Milton Keynes Prison Fire Station	UAV7: Bedford rail station to Ridgmont UAV8: Milton Keynes (MK) Central rail station to Bletchley

Scenario planning	All scenarios	All scenarios	All scenarios	Scenario 2 and 3

Table 4-2 provides the technical parameters for the UAVs used during the simulation scenarios.

Table 4-2 Technical parameters of UAVs used during the simulation scenarios

Parameters of UAVs	Value	Unit
UAV type	Rotary wing	--
Flight time	30	minutes
Cruise speed	90	km/h
Wind resistance	10	m/s

To simulate more complex dynamic airspace, this study simulated a multi-mission scenario between 9:00 am to 12:00 pm for the Bedfordshire area. This study considered the five missions mentioned earlier in this scenario. Three different sub-scenarios were created for each hour of simulation covering the above missions:

1. **First simulation scenario**: This simulation ran between 9:00 am and 10:00 am. In this scenario, all nine NFZs were static without any dynamic obstacles and weather constraints. As a result of this constraint, no UAV could fly over them during this hour. Moreover, this scenario was conducted by 100 UAV trajectories. The railway track inspection mission by UAVs was not considered in this scenario.
2. **Second simulation scenario**: This simulation ran between 10:00 am and 11:00 am. The difference between this and the first scenario was that more complexity was added to the Bedfordshire airspace by considering Railway Infrastructure Monitoring operations and increasing the number of UAV trajectories to 150. The effects of adverse rain and wind were considered in this scenario.
3. **Third simulation scenario**: 200 UAVs' trajectories in airspace were considered in this simulation that ran between 11:00 am and 12:00 pm. In this scenario, airfields were dynamic, while all four recreational areas and

the prison were kept static. Among the four airfields, Luton and Cranfield were available, and therefore, recreational users of UAVs could fly over Luton and Cranfield at some points. This scenario also incorporated severe weather effects. The scheduled inspections of railway tracks by UAV operations were considered in this scenario.

For simplicity, the figure of one of these scenarios is presented in Figure 4-2.

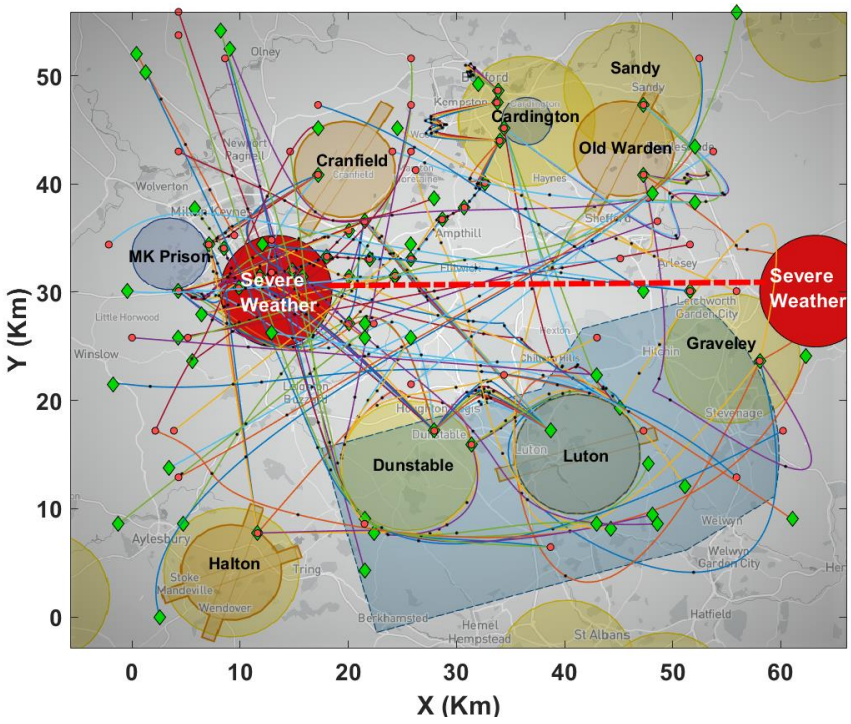

Figure 4-2 Scenario 3: 200 UAVs with extreme weather effects

4.3.2 UAVs States Formulation

With a view to testing the predictive capacity of the proposed model vis-à-vis traffic complexity, the present study used a dataset of simulated historical trajectories that were generated in the previous section. A trajectory is defined as a sequence of UAV states. Each UAV state consisted of the following five elements: longitude (x), latitude (y), timestamps (z), velocity (V), and heading (HA) for each UAV.

These UAV states were used to train datasets to predict UAV traffic flow through the airspace. To predict the airspace flow patterns, UAVs dynamic variables sampled at 10 sec intervals were used to formulate state vector, which showed

the sequential behavior of the UAV. Each UAV's longitude and latitude coordinates at all time instances were generated based on the waypoints of the UAV during the PSO-based simulation scenario. The heading angle and velocity of each UAV can be evaluated using longitudinal and lateral coordinates at each timestamp. Velocity was calculated as:

$$v(k) = \frac{\sqrt{(x(k) - x(k-1)) + (y(k) - y(k-1))}}{z(k) - z(k-1)} \quad \textbf{(4-1)}$$

where k represents current time instant and x and y represent longitudinal and lateral coordinates, respectively. Timestamp has been represented by z in (1). Similarly, heading angle is evaluated using the change of the UAV's trajectory in the x and y coordinate system. Heading angle (HA) can be stated as:

$$HA(k) = \tan^{-1}\left(\frac{y(k+1) - y(k)}{x(k+1) - x(k)}\right) \quad \textbf{(4-2)}$$

Discretized trajectories can be used to evaluate UAV speed, heading angle, and congestion matrices at each time instant. State vector for a certain UAV at a certain time instant can be given as:

$$\text{State} = [\text{Timestamp, Longitude, Latitude, Velocity, Heading Angle}] \quad \textbf{(4-3)}$$

4.3.3 Computation of Complexity Metric: Spatio-Temporal Correlation

The complexity metric deployed in the current study is adapted from a previously published linear dynamic system model applied for ATM domain [261]. The metric identifies a complexity parameter in the vicinity of a UAV for a specified time. Flights that are likely to engage with the reference flight are accommodated via a filter. In this work, a safe distance of 50 m was maintained to avoid any conflict (thus, a drone 50 m from the reference vehicle was omitted from the metric computation as it would not interfere) [262]. The suggested complexity metric was specifically designed to capture, within a reference airspace window, the dynamic behavior of nearby drones. In order to identify the geometric behavior of the trajectories, the velocity of the UAVs ($\dot{U}$) was connected to the position of the UAVs (U) via deployment of the following linear dynamical system [222]:

Let us define matrices P and V as:

$$P = \begin{bmatrix} x_1 & \dots & x_n \\ y_1 & \ddots & y_n \\ z_1 & \dots & z_n \end{bmatrix} \quad and \quad V = \begin{bmatrix} V_{x_1} & \dots & V_{x_n} \\ V_{y_1} & \ddots & V_{y_n} \\ V_{z_1} & \dots & V_{z_n} \end{bmatrix} \tag{4-4}$$

where $x_1, y_1, \text{and } z_1$ denote the spatial-temporal coordinates, and $V_{x_1}, V_{y_1}, \text{and } V_{z_1}$ denote the velocities of the first drone in the vicinity of the reference UAV. The matrix A_U and the vector b are determined with least mean squares:

$$min_{A,b} \|V - (A_U P + b)\|^2 \tag{4-5}$$

The dynamical system defined by A_U and b is the tightest approximation of the current air traffic situation (positions and velocities) by a linear dynamical system. A convergence behavior corresponds to trajectories getting closer to each other, and this increasing proximity is itself associated with heightened complexity. The eigenvalues of the state-transition matrix characterize the behavior of the linear dynamical system and thus can be used to quantify the geometric properties of the considered traffic situation. More precisely, eigenvalues with positive real parts correspond to divergence behavior, while eigenvalues with negative real parts correspond to convergence behavior (Figure 4-3). The complexity is related to convergence behavior since convergence increases the risk of collisions. Consequently, complexity in the vicinity of the reference UAV is defined as follows:

$$c(A_U) = \sum_{Re(\lambda(A_U)) < 0} |Re(\lambda(A_U))| \tag{4-6}$$

The metric $c(A_U)$ is therefore a parameter of the magnitude of the divergence or convergence activity of the dynamic system.

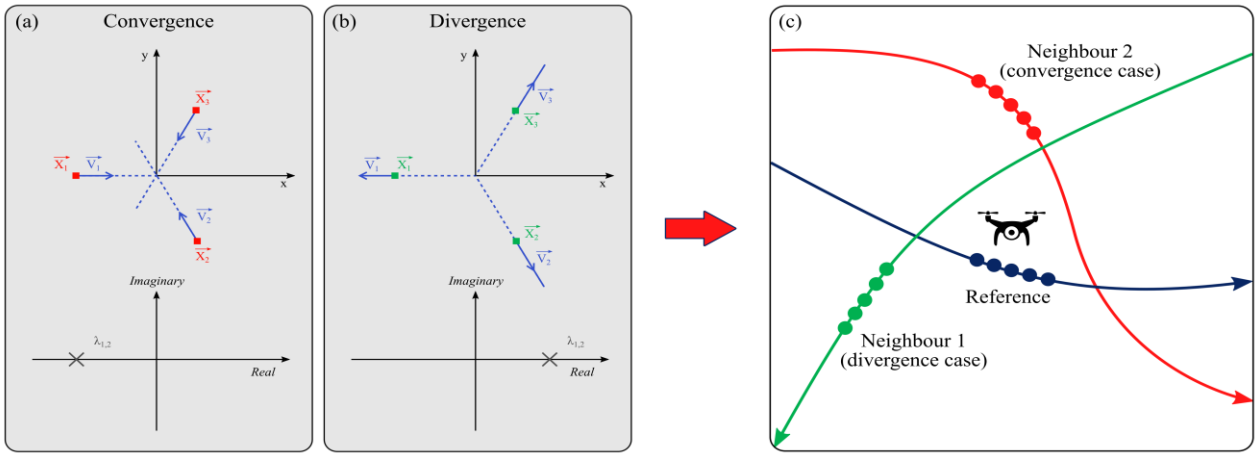

Figure 4-3 The dynamic system and its eigenvalues: for dynamic-system matrix A_U; the eigenvalue loci for two typical situations, namely, convergent (a) and divergent (b). One should note that the real component of the eigenvalues is negative only in the case of convergent trajectories. Meanwhile, the 'neighborhood' of a reference aircraft (for the computation of the complexity metric) is indicated by (c). (Adapted from the work of García et al. [261]**)**

4.3.4 Implementation of Encoder-Decoder LSTM Model

As set out in the introduction, this study aimed to improve the accuracy of complexity prediction in congested airspace. In order to extract complex patterns from time series data, an encoder-decoder architecture was used that relies on LSTM layers and a one-dimensional (1D) convolutional layer. LSTM comprises four regulatory gates, namely, input, forget, cell, and output, in addition to hidden units. Figure 4-4 provides the overall configuration for a single LSTM block, while the mathematical description is as follows:

$$
\begin{aligned}
I^t &= f(W_{ix}x^t + W_{ih}h^{t-1} + W_{ic}C^{t-1} + b_i) \\
F^t &= f\left(W_{fx}x^t + W_{fh}h^{t-1} + W_{fc}C^{t-1} + b_{if}\right) \\
C^t &= F^t \circ C^{t-1} + I^t \circ g(W_{cx}x^t + W_{ch}h^{t-1} + b_c) \\
O^t &= f(W_{ox}x^t + W_{oh}h^{t-1} + W_{oc}C^t + b_o) \\
h^t &= O^t \circ g(C^t)
\end{aligned}
\qquad \textbf{(4-7)}
$$

where the activations of the input, forget, cell and output gates are denoted, respectively, by I^t, F^t, C^t and O^t, and h^t represents the hidden unit. The forecasting time step is represented by the t superscript. The weights matrices are denoted by W, and the bias vectors by b, with corresponding subscripts.

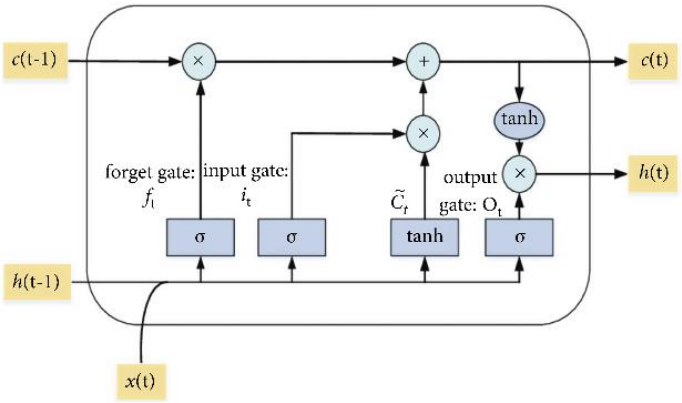

Figure 4-4 LSTM for time series prediction—general architecture [263]

Because input and output sequences may have different lengths, it is difficult to resolve certain problems related to sequence-to-sequence prediction. The encoder-decoder LSTM has been designed to address such problems. The input sequence is encoded into a fixed-length vector by the first model, while the fixed-length vector is decoded by the second model, which also generates an output for the forecast sequence [258]. This configuration is unique due to the use of a fixed dimensional intrinsic representation in the center of the model termed 'sequence embedding' [264]. Both the encoder and decoder may be described as RNNs, i.e., LSTMS.

Figure 4-5 presents the structural diagram for the model proposed in this study. The final dataset is composed of various sizes of simulated trajectories that have developed based on three scenarios outlined in the experiment design and data generation section.

At a given time t, X is used to represent the state vector of the i^{th} trajectory. Subsequently, $\mathrm{X_t}$ is defined as the concatenation of all state vectors(X), which applies to all UAV trajectories in the airspace. The prediction target is complexity matrix $\mathrm{H_t}$. In order to identify neighboring UAVs in such a way that each cube corresponded to an element of $\mathrm{H_t}$, the airspace was divided into small cubes. The complexity metric was then computed for all UAVs currently in each cube. Moreover, $\mathrm{t_{in}}$ represents the duration of the observation window, whilst $\mathrm{t_{pred}}$ represents the prediction horizon.

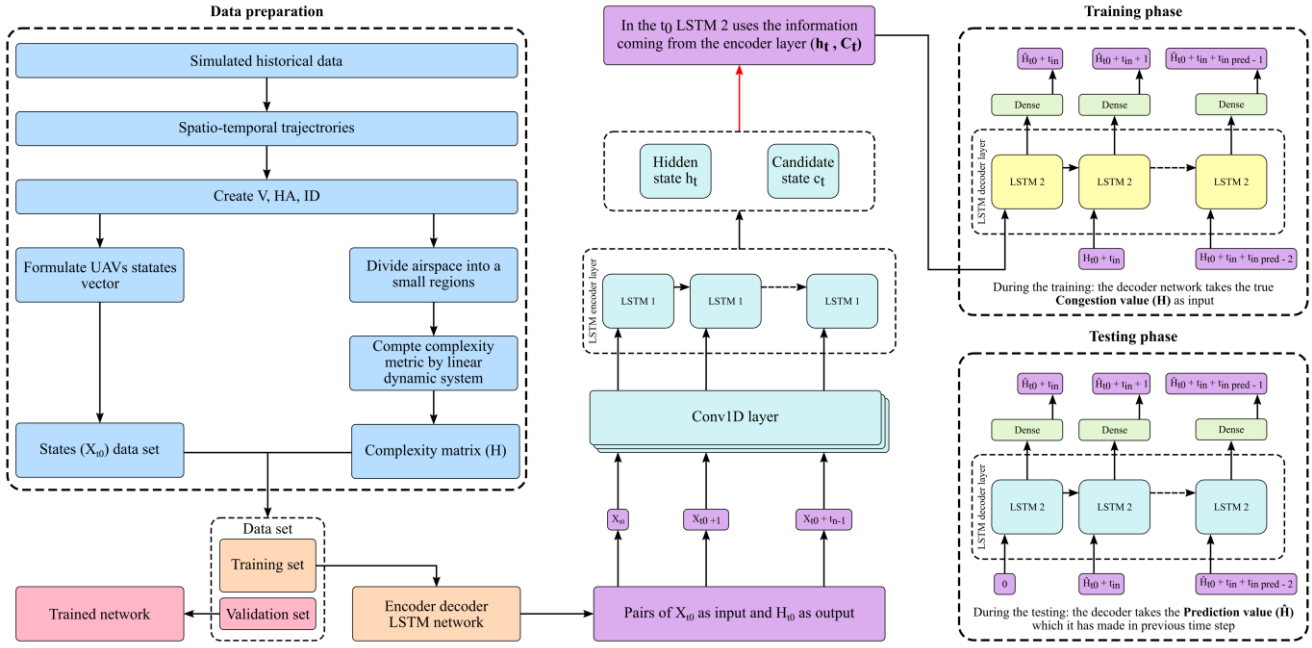

Figure 4-5 Overview of the proposition for the UTM traffic flow prediction model

The dataset was composed of pairs of sequences $(\mathrm{X_{t_0}}, \mathrm{X_{t_0+1}}, \ldots \mathrm{X_{t_0+t_{in-1}}}), (\mathrm{H_{t_{0+t_{in}}}}, \mathrm{H_{t_0+1}}, \ldots \mathrm{H_{t_0+t_{pred-1}}})$. For complexity prediction purposes, an encoder-decoder LSTM model was employed. The encoder network used a sequence of UAV states $(\mathrm{X}_{t0} \ldots \mathrm{X_{t_0+t_{in-1}}})$ to output an encoding vector. The decoder network then uses the encoding vector and the complexity matrix predicted at the last timestep $\widehat{H}_{t-1}$ to predict the next complexity matrix $\widehat{H}_t$. The first layers of the encoder network can be designated a 'single-dimension convolutional layer'. These layers process the input sequence corresponding to the UAV states. Different f filters comprise each layer, and these filters further encompass a kernel of learnable value parameters, with the dimension $d * m$, where m and d indicate the input sequence, and kernel width, respectively. The result of the input sequence, as it is relating to the distance, is aggregated by the individual filters, while each output series element comprises data from several serial time increments. This can be represented as follows:

$$y_{ij} = \emptyset\left(\sum_{k=-d}^{d}\sum_{l} w_{kl}^{j} x_i + k, l\right) \qquad \textbf{(4-8)}$$

where y_{ij} denotes the j^{th} vector element of the i^{th} sequence term, and x_{kl} signifies the l^{th} vector element of the k^{th} sequence term of the input. Meanwhile, w_{jkl} represents the k^{th} weight of the kernel of the j^{th} filter, as related to the l^{th} dimension of the input sequence; finally, Φ denotes an activation function. The implementation of convolution operations occurs exclusively along the time dimension of the input sequence hence, the use of the term 'one-dimensional' (1D). The convolutional layers were deployed to identify dependencies within a short time period. Thus, the processing of the LSTM layer is facilitated by incorporating dynamical information from the preceding and subsequent time steps. Without the convolutional layer, conversely, the LSTM layers would merely be capable of processing each of the time intervals consecutively, absent any data from subsequent time increments. The parameters of the proposed model are presented in Table 4-3 below.

Table 4-3 List of the proposed model's parameters

Parameter	Description
ID	A unique code is assigned to a single aircraft UAV to identify flight mission
P	The UAV mission priority
t	The timestep when the UAV is passing the waypoint
V	UAV Velocity
HA	Heading Angle
UAV States	State = [Timestamp, Longitude, Latitude, Velocity, Heading Angle]
Complexity Metric $\mathrm{c}(\mathrm{A_U})$	The complexity metric is a linear dynamic system model that identifies a complexity parameter in the vicinity of a UAV for a specified period of time.

4.3.5 Model Training

The objective of this supervised learning model is to provide accurate congestion forecasting in UTM systems. For the model's objective to be realized, it is necessary to construct a training set by deploying training input/output pairings. The input vector $\mathrm{X}_{\boldsymbol{t}}$ comprised sequences of UAV states, duly matched with complexity value sequences for the airspace as a whole. To generate the training outputs, for each time increment t, the complexity matrix H_t is defined as an $n \times n$. The hidden states of the final LSTM layer form the encoding vector—no other encoder data will be supplied to the decoder network; the decoder network is

composed of several LSTM layers whose hidden states are initialized with the encoding vector. These LSTM layers are followed by several dense layers. The output from these layers is a vector that had an n^2 dimension, equivalent to the matrix of the complexity metric.

The decoder input is formed by two elements, i.e., the encoder output, and the previously predicted or decoded output sequence term. During the training, the decoder network is trained by a method referred to as 'Teacher forcing' [265], [266]. This method consists of using the output predicted at the previous time step as input for the decoder network. The prediction identified at the preceding time interval is thus accounted for within the inference mechanism.

The process of the proposed encoder-decoder LSTM model and data sliding window is illustrated in Figure 4-6. The encoder network (on the left) uses a sequence of UAV states to compute an encoding sequence, while the decoder network (on the right) uses this encoding sequence and the complexity matrix output at the last timestep. The duration of the observation window (i.e., the number of timesteps observed before making predictions) is 36 timesteps which corresponds to three minutes, or 180 s, since one time step comprises five seconds. The prediction is performed at any given time t_k, using the last 3 minutes of traffic as input (from t_k - 3min to t_k), and predicting the next 3 minutes (from t_k to t_k + 3 min).

The list of model hyperparameters is shown in **Error! Reference source not found.**.

Table 4-4 List of the proposed model's hyperparameters values

Parameter	Value
Batch size	128
(1-D) kernel width d	3
(1-D) filter f	512
Hidden layers (LSTM)	128
Activation	ReLU
Optimizer	Adam optimizer [267]
Learning rate	0.001
Epochs	500
Loss function	RMSE

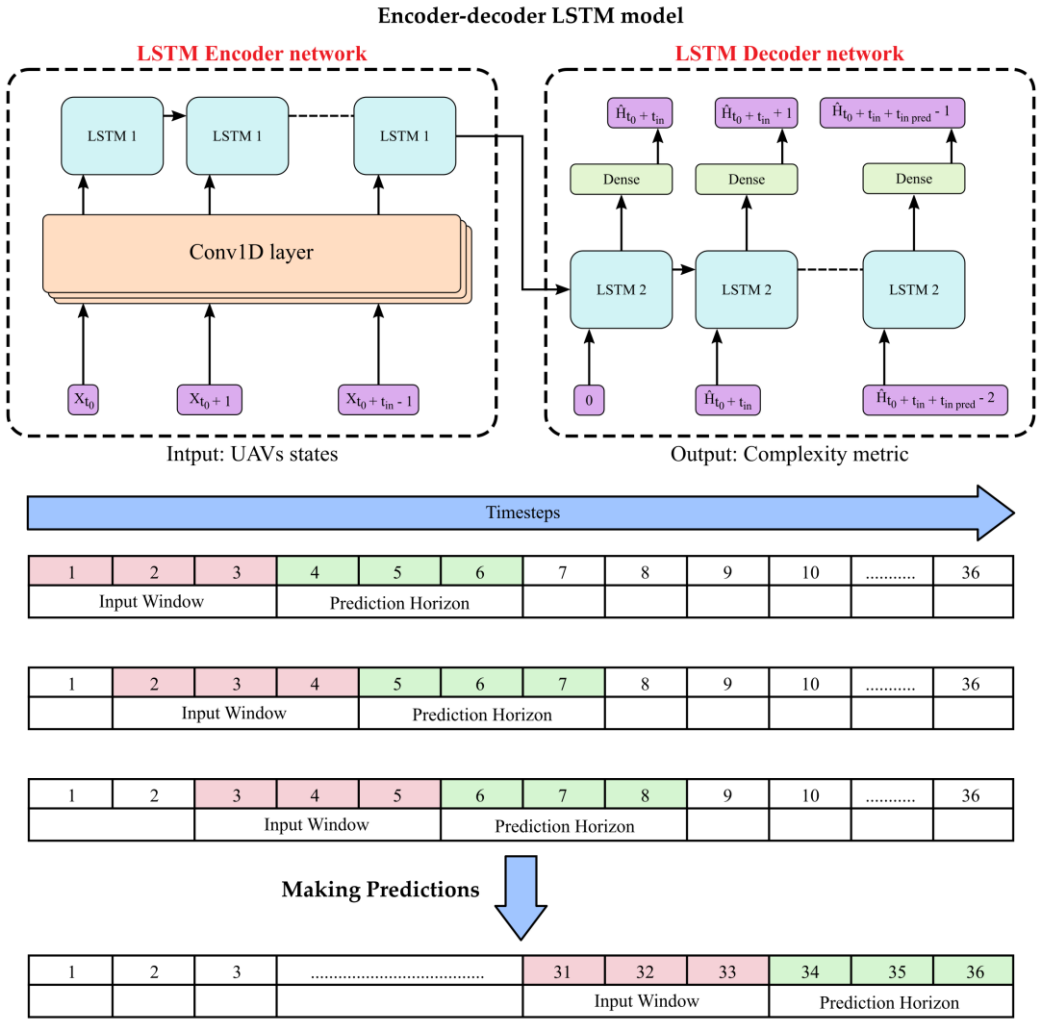

Figure 4-6 The process of data sliding window for creating a time series data

The root mean square error (RMSE) was used to assess the performance of each selected approach and served as the estimation criterion for the performance of the proposed model. The following expresses the RMSE, which denotes the loss function:

$$\mathrm{RMSE} = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(C_{ac} - C_{pr}\right)^2} \qquad \textbf{(4-9)}$$

where N denotes the total number of airspace (x, y) coordinates, C_{ac} signifies actual congestion volumes at any specific point k, and finally, C_{pr} represents the forecast congestion at the precise (x, y) coordinate. Since differences between

forecast and actual values are ‘less’, higher predictive accuracy is reflected by a lower RMSE value.

4.4 Results and Discussion

4.4.1 Prediction Result

4.4.1.1 First Scenario Simulation

This case postulated a scenario of 100 UAVs with an NFZ but no adverse weather conditions; the objective was to evaluate airspace congestion. Figure 4-7 presents a heat map for the scenario congestion matrices, and it addresses the Bedfordshire airspace congestion caused by UAVs in the neighborhood. The airspace was analyzed in order to predict the density of the UAV trajectories at each square km area. The low to high densities of UAVs per square km are represented by red and green, signifying the highest and lowest complexity, respectively.

The figure illustrates that most of the highly complex regions in the future timestamp were captured by the proposed model. The RMSE value of the prediction was employed to evaluate the performance of the LSTM trained architecture. RMSE reached a value of 0.1662 for this scenario, indicating high model accuracy.

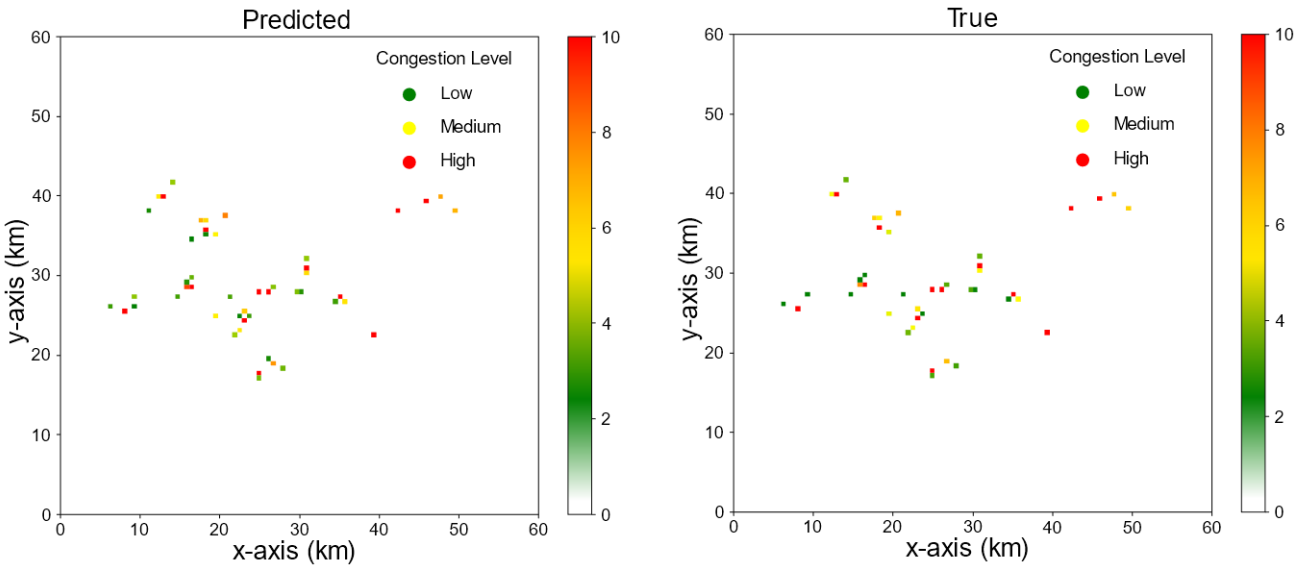

Figure 4-7 First scenario: Actual and predicted complexity in the airspace

Furthermore, it is evident from the residual plot (Figure 4-8) that predicted complexity was very close to true complexity, with very few wrong detections during the prediction process. Moreover, although most of the congestion areas were captured, the regions with high complexity depicted a small difference in magnitude between true and predicted complexity.

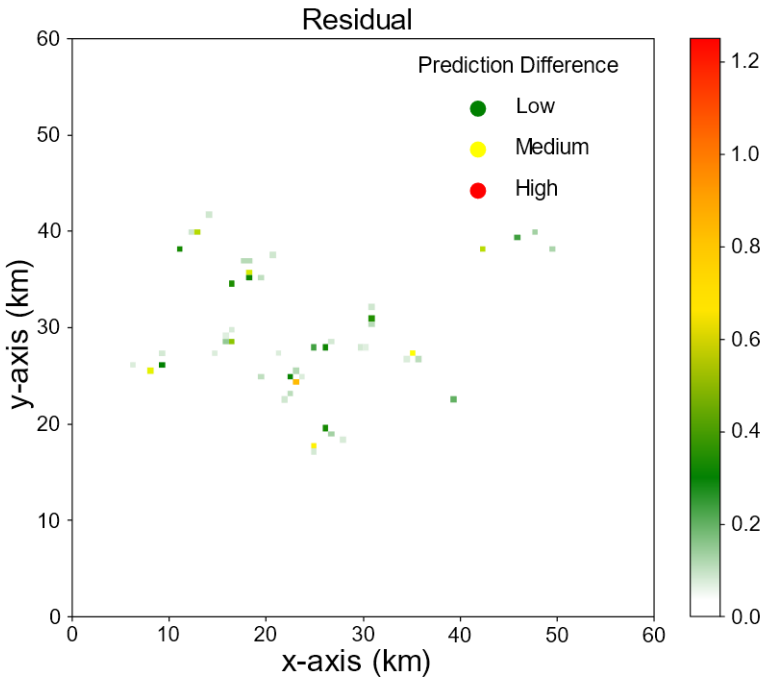

Figure 4-8 The difference between predicted and true congestion—Scenario 1

Statistical analysis of the actual, predicted, and residual complexity figures was completed to present the above information in greater depth. In this regard, the analysis normalized the complexity values for each complexity point between 0–100 whilst also taking into account the maximum complexity points for all true, predicted, and residual vectors. The percentages for the complexity figures were then plotted against the trajectory points. The largest value of residual complexity percentage represents the maximum dissimilarity between empirical truth and predicted complexity. Furthermore, the number of residual complexity points below an acceptable threshold were counted to provide a means of evaluating the performance of our proposed methodology. The statistical analysis of the residual complexity for scenario 1 is presented in Figure 4-9 below.

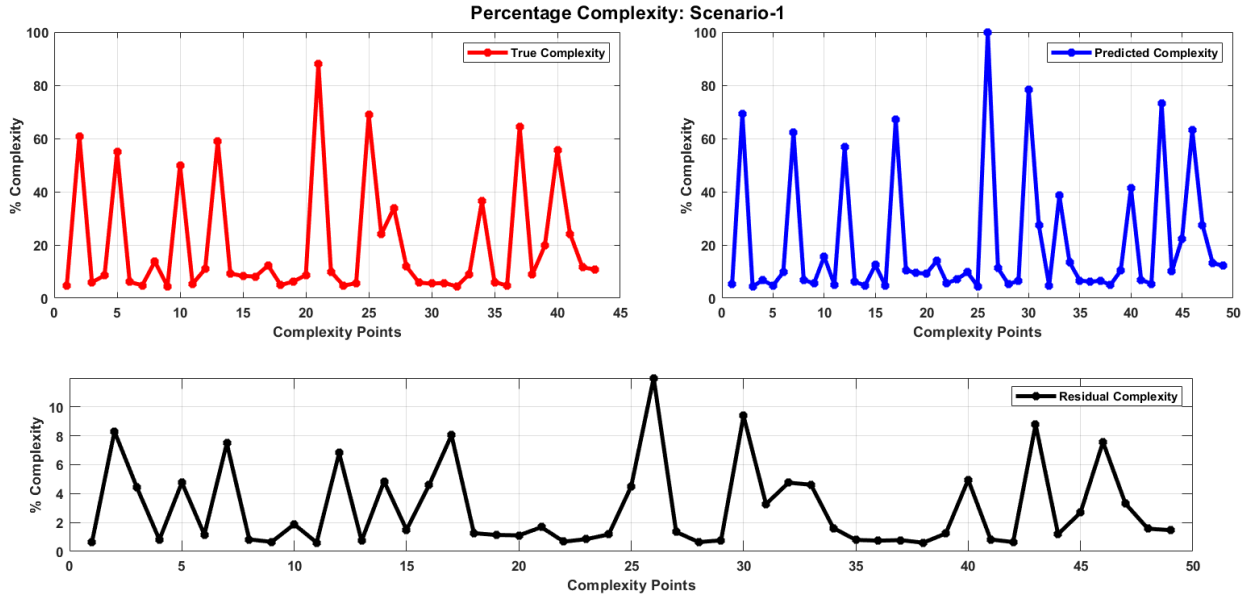

Figure 4-9 Percentage residual complexity—Scenario 1

The statistical analysis of percentage complexity for scenario 1 showed that the maximum dissimilarity in complexity between ground truth and prediction using LSTM was 12%. Furthermore, 83% of complexity points had less than 5% residual complexity, which indicates better prediction performance. The peak residual complexity values correspond to the regions with the highest complexity trends.

4.4.1.2 Second Scenario Simulation

In scenario 2, the putative complexity of the Bedfordshire airspace environment was increased. The number of UAVs increased to reach 150 by considering railway infrastructure monitoring operations, dynamic recreational areas, and uncertain weather with adverse wind and rain. Figure 4-10 comprises the resulting congestion heatmap. As the heatmap demonstrates, by observing the speed, congestion areas, and dynamic behavior of the UAVs, the encoder-decoder model captured major complexities within the airspace. The performance of this model may be quantified via the RMSE (0.2852) for training and validation at each epoch. Similarly, Figure 4-11 represents the residuals of the predicted values in relation to the actual ones. Density increases in line with

increases in the number of UAVs, as the figure clearly indicates; this phenomenon stems mainly from significant fluctuations in observable traffic patterns. Increasing density, furthermore, is driven to a large extent by dynamic weather constraints. The residual in the previous scenario was calculated to observe the non-detection, wrong detection, and detection differences in the true and predicter airspace complexity values. Additionally, the residual plot shows a very small difference between the actual and true complexity heat maps.

Moreover, the residual analysis in Figure 4-12 demonstrates that the maximum dissimilarity of complexity between ground truth and prediction using LSTM was 43%. Moreover, 70% of complexity points had less than 10% residual complexity, indicating better prediction performance. The peak residual complexity values correspond to regions with maximum complexity trends.

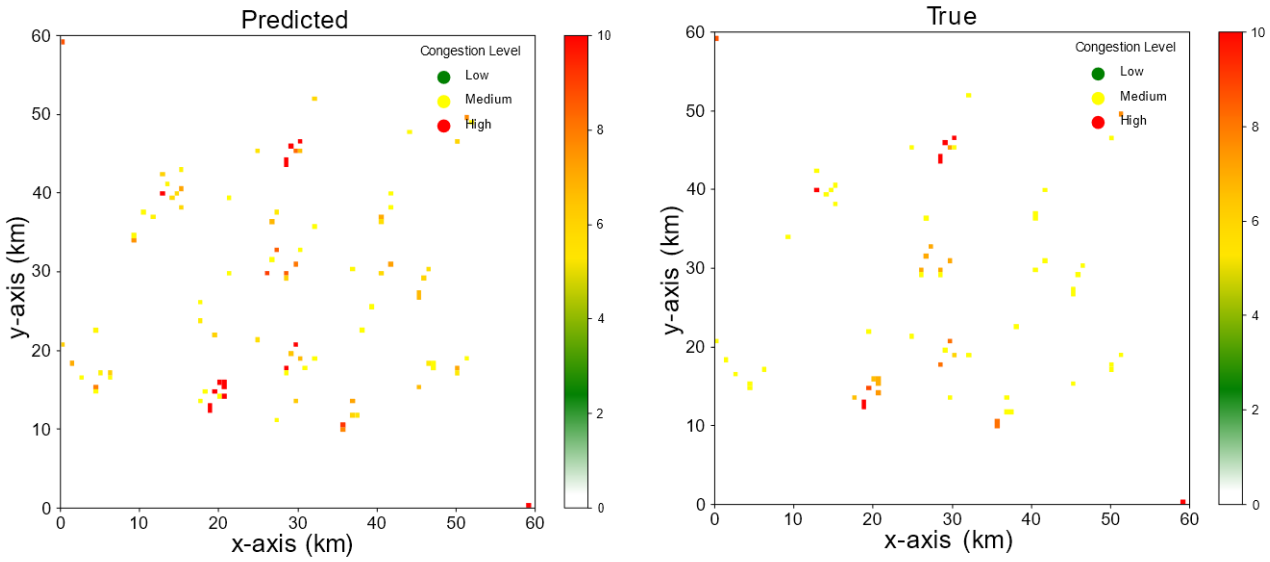

Figure 4-10 Scenario 2: Actual and predicted complexity in the airspace

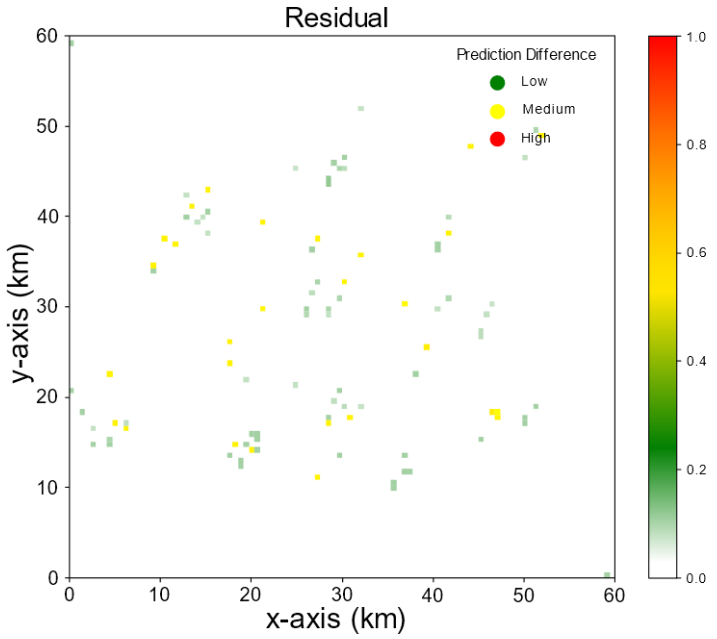

Figure 4-11 The difference between predicted and true congestion—Scenario 2

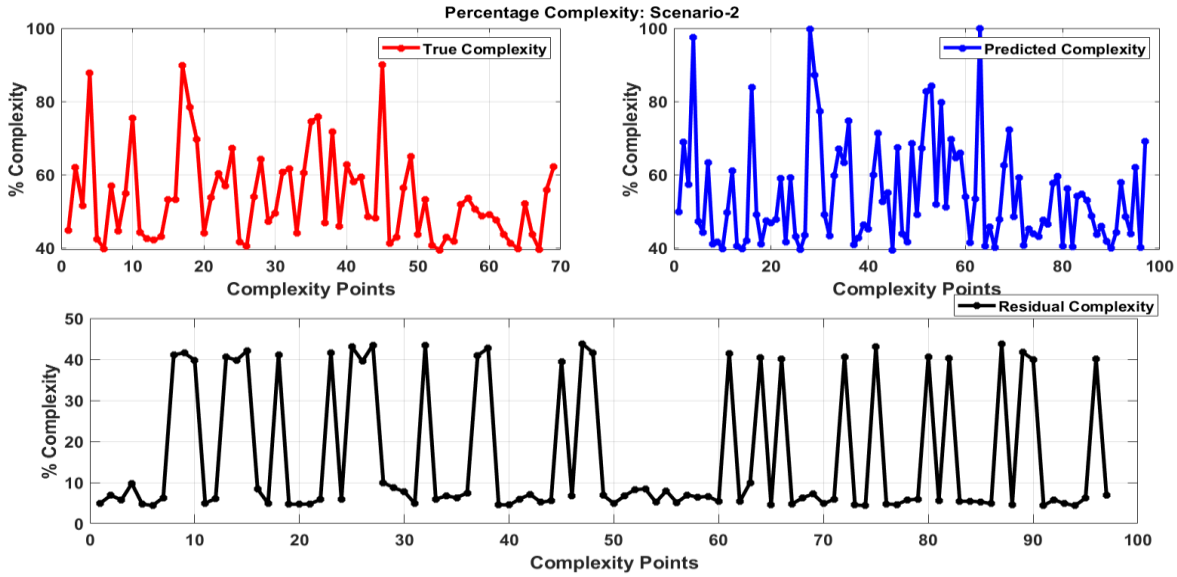

Figure 4-12 Percentage residual complexity—Scenario 2

4.4.1.3 Third Scenario Simulation

Altogether, 200 UAV flights in airspace were considered in this scenario, including the scheduled inspections of railway tracks by UAV operations. Although the airfields were dynamic, all four recreational areas and the prison remained static. Furthermore, severe weather effects were considered during scenario 3. Figure

4-13 provides the resulting congestion heatmap, indicating that the temporal and spatial transition patterns of air traffic flight flow were indeed learned by the proposed model. Future air traffic flows may be predicted since the primary characteristics of these flows are captured. The third scenario generated an RMSE value of (0.6021), and the prediction from the trained model architecture is sufficiently close to the true congestion value. This indicates that the proposed ATFP architecture performs optimally.

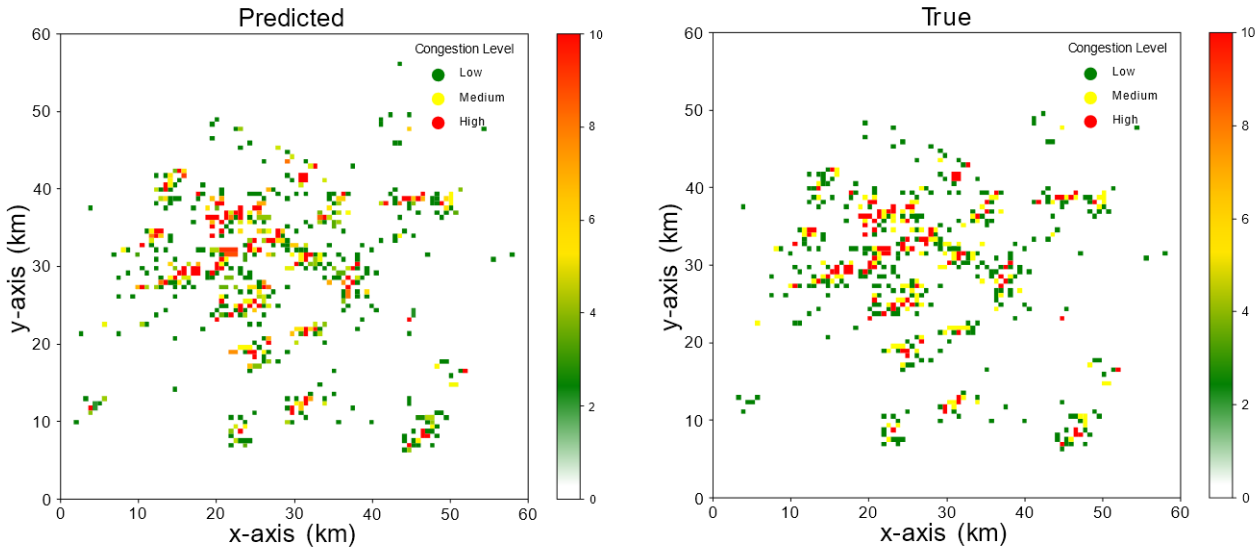

Figure 4-13 Scenario 3: Actual and predicted complexity in the airspace

A residual plot is presented in Figure 4-14 to highlight the accuracy of predictions made by the proposed LSTM network. Given the higher density of UAVs in airspace, the prediction also depreciated, which is evident from the absolute difference between true and predicted airspace congestion. Moreover, given the incidence of incorrect detection, some high congestion spots were visible in the residual plot. Most of the predicted airspace complexity values were correct, which is evident from the RMSE between true and predicted airspace complexity. It was observed from Figure 4-15 that the maximum dissimilarity of complexity between ground truth and prediction using LSTM was 10%. Moreover, 97% of complexity points had less than 5% of residual complexity, which indicates optimal prediction performance. The peak residual complexity values correspond to regions with maximum complexity trends.

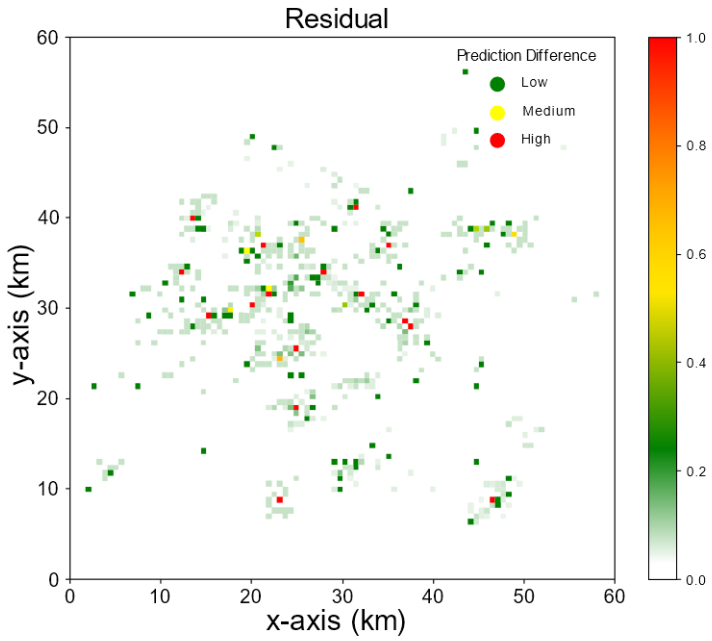

Figure 4-14 The difference between predicted and true congestion—Scenario 3

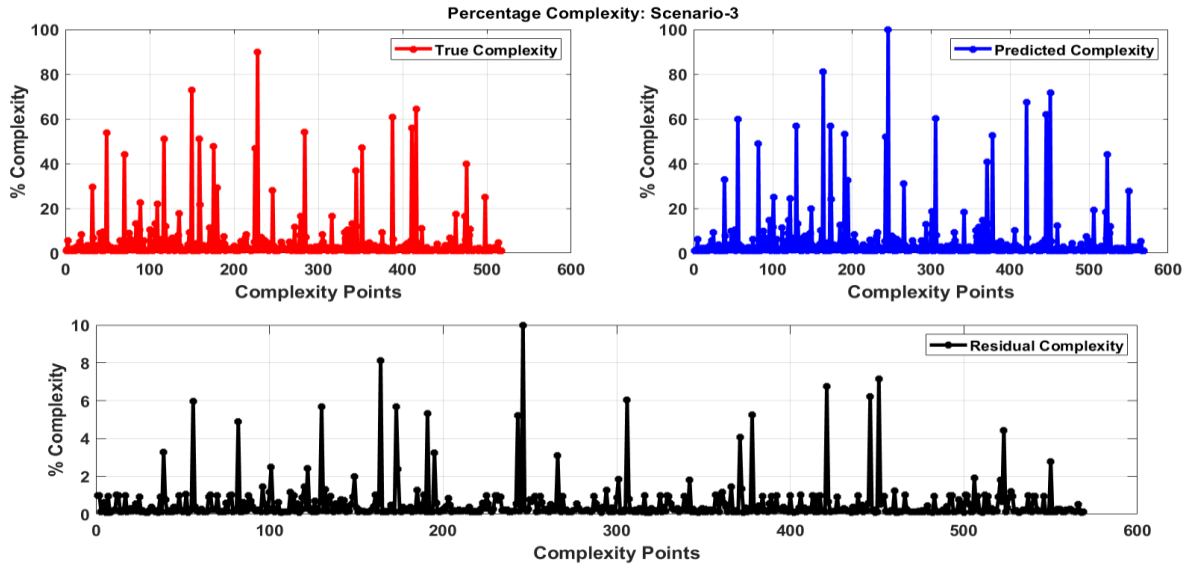

Figure 4-15 Percentage residual complexity—Scenario 3

4.4.2 Justification of the Selection of the Look-Ahead for Prediction

The choice of a 3-minute historical sample window and look-ahead time is dictated by the specific requirements of the applications. In the context of drone delivery services, emergency operations, and inspection-related UAV tasks with mission durations of less than 20 minutes, a short historical sample window and

a brief look-ahead time are appropriate. A historical sample window ranging from 3 to 10 minutes is sufficient to capture recent changes in traffic patterns and weather conditions. Similarly, a look-ahead time of 3 to 10 minutes is enough for providing timely predictions that can be used to plan a safe flight trajectory in advance, enabling the rerouting of drones or adjustment of delivery schedules. However, the optimal value for these temporal parameters depends on the characteristics of the data and the complexity of the prediction problem and requires experimentation and tuning of the deep learning model.

To address this, a sensitivity analysis was conducted to examine the effects of longer and shorter look-ahead times and input windows. The objective of this analysis is to validate the prediction architecture using a simple predefined time series (e.g., 1 minute as the input window and look-ahead time). Additionally, the analysis aims to assess the impact of these temporal parameters on the accuracy and stability of the congestion prediction model, providing valuable insights into the optimal parameter values within the given context. For simplicity, Figure 4-16 below showcases the analysis results for a scenario involving 100 UAVs with a no-fly zone (NFZ) but without adverse weather conditions (as detailed in section 4.3.1). Different input sample windows and look-ahead times are utilized for predicting traffic congestion in UTM. For example, the notation (1:1) in Figure 4-16 corresponds to a 1-minute input sample window and a 1-minute look-ahead time. This implies that the prediction model considers a historical data window of 1 minute as the input to make predictions, while also looking ahead by 1 minute to forecast traffic congestion. The computation time indicated in the figure represents the duration required to compute the predictions. The RMSE is used to measure the accuracy of models. A lower RMSE indicates better accuracy, as it represents a smaller average difference between the predicted and actual values.

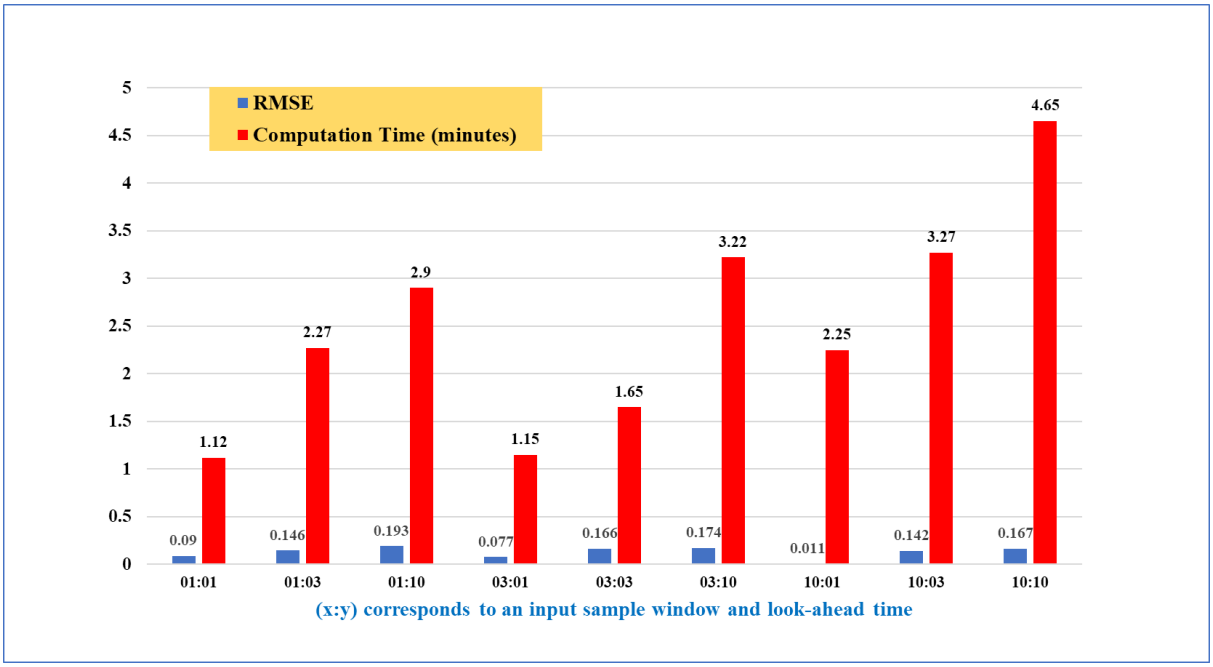

Figure 4-16 Comparison of Input Sample Windows and Look-Ahead Times for Predicting Traffic Congestion in UTM

In the above figure, the vertical blue bar represents the prediction error, while the red bars depict the computational complexity. From the above figure, it is evident that the prediction error is the lowest (0.011) for a 10-minute input sample window and 1 minute look-ahead time. Conversely, the prediction error is the highest (0.193) for a 1-minute input sample window and 10-minute look-ahead time. Moreover, the computation time is the lowest (1.12 and 1.15) for 1 min and a 3-minute input sample window and 1 and 1-minute look-ahead time respectively, and highest (4.65) for a 10-minute input sample window and 10-minute look-ahead time.

When the time input window is shorter, the model has access to less historical information and context for making predictions over a longer look-ahead period. This limited context can introduce increased uncertainty and potentially higher prediction errors. Conversely, using a longer input window provides the model with a more extensive historical context, enabling it to capture additional patterns and dependencies in the data. This expanded context may result in more accurate predictions, leading to lower error rates. In this study, the historical data is constrained to a time range of 20 minutes, which limits the achievable look-

ahead time. To assess the impact of extending the look-ahead time, experiments were conducted using a longer look-ahead time of 10 minutes. Notably, the analysis revealed that increasing the look-ahead time did not yield a significant increase in the (RMSE) metric.

To establish a standardized experimental condition, the look-ahead time was set to 1 minute, allowing for an investigation into the impact of modifying the input sample window. Referring to Figure 4-16, the following trends can be observed:

With a 1-minute input sample window, the RMSE is recorded at 0.09, and the computational time is 1.12.

When the input sample window is extended to 3 minutes, the RMSE improves to 0.07, accompanied by a computational time of 1.15.

Further increasing the input sample window to 10 minutes results in a substantial reduction in RMSE, which decreases to 0.01, albeit with a longer computational time of 2.25.

It is noteworthy that as the duration of the sample window is increased, a trade-off is observed: the computational time increases, while simultaneously, the prediction error decreases.

In the context of the specific application use-cases examined in this study, it has been determined that accurate predictions can be achieved while maintaining efficient computation time with a 3-minute look-ahead. Figure 4-16 demonstrates that the fastest computation time is attained when utilizing a 3-minute input window, while (RMSE) remains competitive compared to larger input windows. Extending the look-ahead time involves a trade-off between improved safety and efficiency versus increased complexity and reduced prediction accuracy. In highly automated systems like UTM, longer look-ahead times are possible but introduce more complexity and data challenges (i.e., the scarcity of historical UTM data). Obtaining precise predictions for distant future points becomes harder, and it requires additional computational resources. The dynamic UAV environment may also change rapidly, impacting the accuracy of long-term predictions. This consideration is crucial for UTM operations, where minimizing complexity and

computational demands while maintaining accuracy is essential, particularly in resource-constrained scenarios like emergency operations.

4.4.3 Discussion on the Generalization of the Proposed Model

The combination of convolutional and LSTM networks in the proposed model creates a strong foundation for addressing large-scale traffic prediction problems while remaining trainable on resource-constrained devices. By incorporating convolutional layers, the model can effectively capture spatial patterns and relationships within the traffic data. This enables it to extract relevant features and understand the complex dynamics of traffic flow in large-scale scenarios. Additionally, the inclusion of LSTM networks allows the model to capture temporal dependencies and long-term patterns in the data. LSTM networks are specifically designed to handle sequential data, making them well-suited for capturing the time-varying nature of traffic patterns.

The proposed model underwent initial training and testing using a dataset consisting of three distinct scenarios, and the implementation details and results are published in [268]. To assess the proposed model's generalization ability, it is essential to test it with unseen datasets. This evaluation will determine the model's effectiveness in handling various air traffic congestion situations and its ability to generalize beyond the training data. These new scenarios incorporate various factors, such as dynamic weather conditions, increased UAV traffic, and evolving airspace restrictions, as outlined in the experiment design and data generation section (section 4.3.1). For this purpose, the following datasets are used.

In the first scenario with 100 UAVs, there are a total of 132,148 data points. In the second scenario with 150 UAVs, the dataset reaching 181,505 data points. Finally, during the third hour of simulation with 200 UAVs, the dataset contains 252,129 data points. The proposed model is trained with the help of data from scenario 1 that only incorporates NFZ and then validated with the data from scenarios 2 and 3 that are completely unseen as they include random factors such as dynamic airfields, adverse wind and extreme weather conditions. The proposed model achieved an RMSE of 0.16 on the training set (scenario 1). The

model's performance was evaluated on validation sets (scenarios 2 and 3) that were not included in the training data, resulting in RMSE values of 0.28 and 0.60 respectively. Additionally, a Normalized RMSE (NRMSE) performance metric was measured to assess the accuracy of a model's predictions, considering the actual and predicted results obtained in the aforementioned three scenarios. NRMSE takes into account the range (max-min) of the datasets along with RMSE, providing a more comprehensive evaluation of model generalization and suitability under dynamic constraints, such as changing weather conditions and dynamic airfields. The formula for NRMSE can be expressed as:

$$\mathrm{NRMSE}(\%) == \frac{\sqrt{\sum_{i=1}^{n}(y_i - \hat{y})^2}}{\max(y) - \min(y)} \times 100 \qquad \textbf{(4-10)}$$

The results of this NRMSE (%) is presented in Figure 4-17.

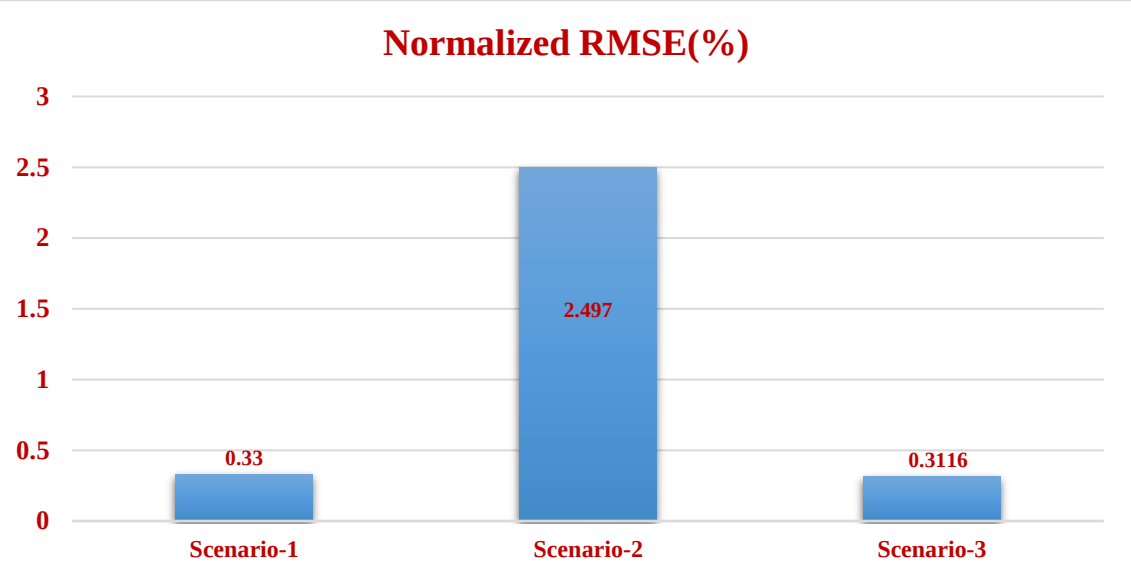

Figure 4-17 Normalized RMSE on training data (Scenario 1) and validation data (Scenarios 2 and 3)

According to Figure 4-17 , the NRMSE (%) is less than 3% for all scenarios. This shows that the model's is able to generalize even under unforeseen and varying

conditions such as dynamic obstacles, emergency operations, and weather changes. However, it is important to acknowledge that there are some limitations in this implementation. Firstly, the challenges posed by the lack of available real-world UTM datasets. The absence of such datasets limits the ability to identify optimal models and fully validate the proposed approach. While a diverse range of scenarios has been simulated, the inclusion of real-world data would provide more accurate representations of actual air traffic dynamics. Secondly, this work focuses on a short-term input sequence, such as data from the past 3 minutes. Failing to account for long-term sequences inhibits the deep learning model's ability to generalize traffic patterns well and can limit prediction accuracy. Furthermore, it is worth noting that the scope of this study was limited to a single airspace in Bedfordshire, UK. While this constraint may limit the generalizability of the findings to a broader context, it serves as a stepping stone for future research. The proposed model architecture can be adapted and applied to larger cities with larger-scale air traffic flows. By scaling up the study to encompass more extensive and diverse airspace, the model's performance can be further evaluated and validated. Finally, the implementation of convolutional and LSTM networks in the proposed model enables effective generalization in large-scale traffic prediction. However, it is important to consider a limitation: the model is classified as a "black box" due to its internal workings and lack of interpretability. This hinders understanding, discourages deployment, hampers debugging and identifying performance issues, and restricts further model analysis and improvement. Addressing this limitation is crucial for enhancing trust, transparency, and practical applicability in traffic prediction.

4.4.4 Comparison Between Existing Approaches and The Proposed Model

As noted earlier, the predictive quality of the proposed model has been compared with two other predictive methods, as described in Section 4.2. The three scenarios were used to assess the performance of each approach. For analytical comparison with the proposed model, the first selected architecture was a shallow NN-based model (with dense network connections) as described in [269]. The

second was a regression architecture-based Nonlinear Auto Regression, with External input (NARX) model, as described in [236]. Calculation of the RMSE was used to assess the performance of each selected approach, and RMSE served as the estimation criterion for the performance of the proposed model. The predictive quality of the proposed LSTM, meanwhile, was evaluated via a comparison of the descriptive statistic values of the predictions with the actual values. Between the actual and predicted values, the descriptive statistics were as follows:

$$\text{Mean } (\mu) = \frac{1}{N}\sum_{k=1}^{N}\left|C_{ac} - C_{pr}\right| \quad \textbf{(4-11)}$$

$$\text{Standard deviation } (\delta) = \sqrt{\frac{1}{N}\sum_{k=1}^{N}\left(\left(C_{ac} - C_{pr}\right) - \mu\right)^2} \quad \textbf{(4-12)}$$

$$\text{AMPE } (\eta) = \frac{1}{N}\sum_{k=1}^{N}\left|\frac{(C_{ac} - C_{pr})}{C_{ac}}\right| \times 100\% \quad \textbf{(4-13)}$$

where N signifies the overall number of airspace trajectories (x, y). Meanwhile, C_{ac} denotes true congestion values, as identified by using the dynamic linear UAV model. C_{pr} signifies the forecast congestion at the particular airspace coordinates (x, y), based on the LSTM encoder-decoder architecture. The degree to which the predicted point deviates from the data mean is assessed via the standard deviation. In order to present the total deviation of the predicted values from the mean value, every predicted point variation is measured and summarized. The prediction accuracy of the model, in percentage terms, is measured via the absolute mean percentage error (AMPE). Figure 4-18 illustrates the descriptive statistics and RMSE calculations for the actual and predicted values.

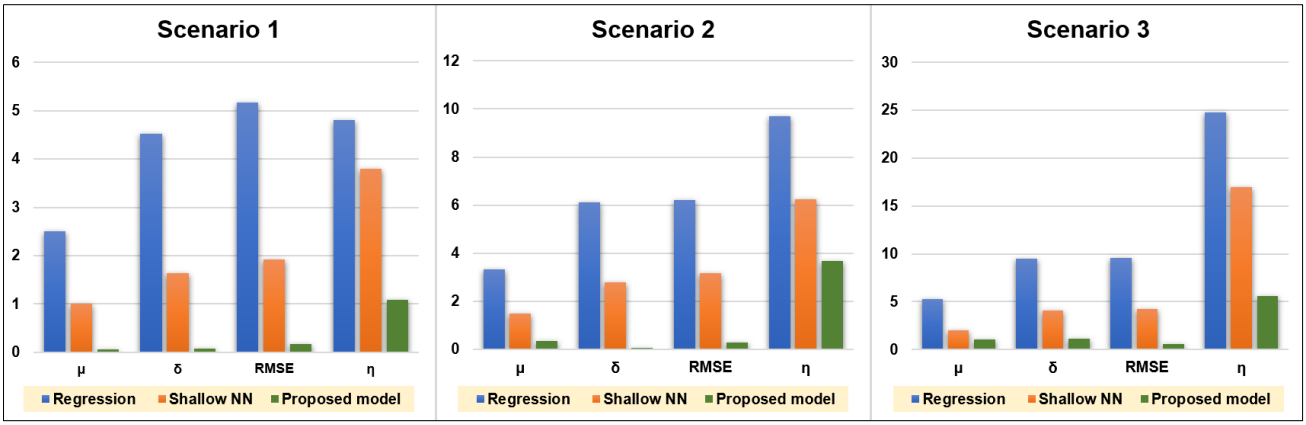

Figure 4-18 Comparison of different prediction approaches

As the graphs indicate, the LSTM model has the smallest RMSE value of all the indices. This confirms the hypothesis that combining the advantages and features of various algorithms and models (i.e., 1D convolutional and LSTM) affords superior problem resolution [270]. Clearly, this also applies to the problem of ATFP in Bedfordshire considered in this study. The results of the comparison demonstrate a superior performance of the deep learning model compared to the existing approach. Toolbox availability renders Shallow and Regression networks easy to construct, but these fail to capture the complicated relationship between complexity metrics and states.

The superiority of the encoder-decoder LSTM predictive model stems not merely from its predictive capacity, but also from its input and output structures, which facilitate airspace complexity predictions with consecutive time steps. When airspace congestion is presented via a temporal and spatial scale, airspace complexity can be assessed more intuitively. UTM operators may thus look to the future with a sense of how airspace complexity will vary with regard to different time slots. The proposed model shows superior predictive performance in comparison with the other approaches, because both temporal and spatial dependencies are modelled. The three scenarios cited above also simulate differing meteorological ambiguities. Only the influence of historical data (temporal, but not spatial correlations) is reflected in the regression model. The latter is thus notably dependent on the magnitude of the data and the look-ahead horizon. Unexpected variations in air traffic flow, weather conditions, and flow

patterns may also impact the performance of the regression model. Air traffic complexity is more effectively captured by the shallow NN (used in conjunction with a dense network connection). The spatial coordinates of airspace complexity could be predicted more accurately than with the regression model. Nonetheless, the proposed model also demonstrates superior performance in comparison with the shallow NN.

4.4.5 Comparison Between Existing Approaches and The Proposed Model: Performance vs. Time

The prediction performance parameters narrated above have been plotted against prediction time in order to temporally and spatially visualize the efficiency of the proposed methodology. Figure 4-19 shows that complexity prediction performance parameters (such as RMSE and AMPE have much smaller values in the proposed model than other already-proven prediction approaches (such as shallow NN and regression model (NARX)). It can also be seen that the AMPE grows much faster for adverse wind, rain, and extreme weather front scenarios between 10:00 AM to 12:00 PM.

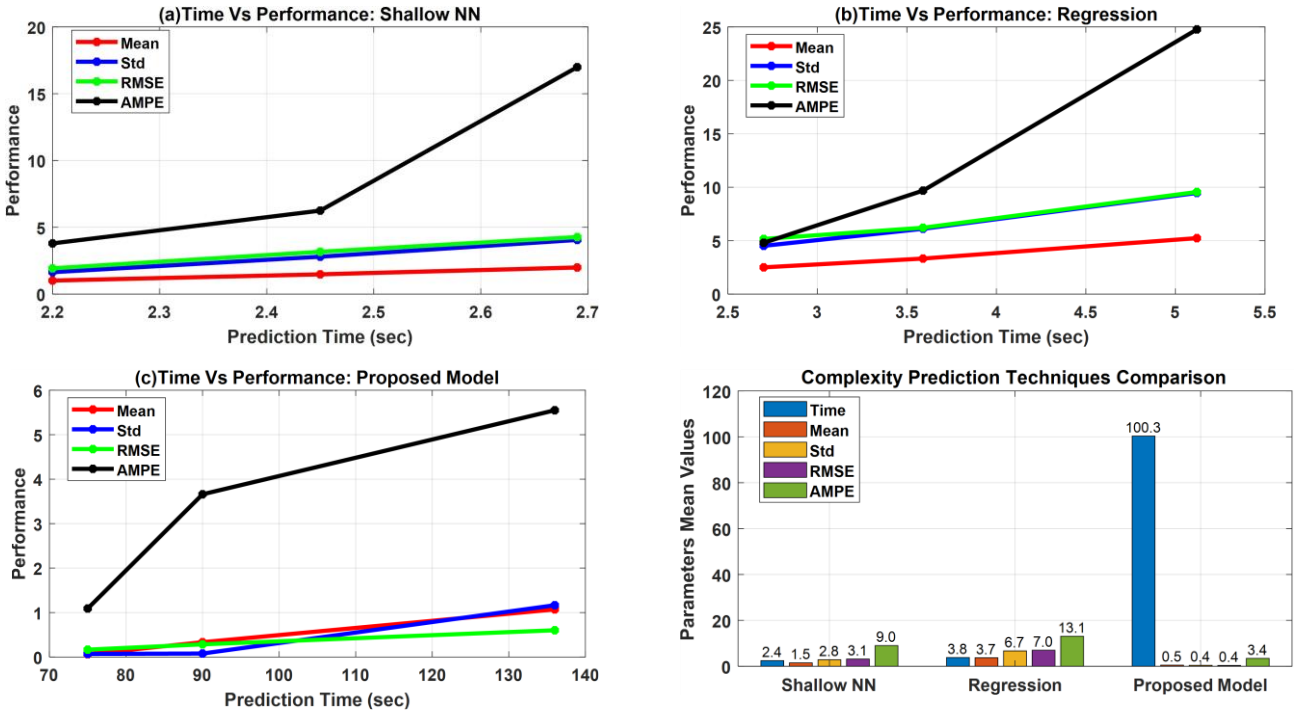

Figure 4-19 Prediction vs. time for (a) shallow NN, (b) regression, and (c) proposed model

This study analyzed the mean values of the computation time and performance parameters across the temporal domain between 9:00 AM and 12:00 PM (scenario 1 to scenario 3). This is presented in the bar chart displayed in Figure 4-19. Additionally, the mean value analysis shows that shallow NN had steady performance on the computation side with an average time of 2.4 s, although it displayed poor statistical performance with RMSE:3.1 and AMPE:9.0. The regression had slightly longer average computation times of 3.8 s and worse complexity prediction statistics with RMSE:7.0 and AMPE:13.1. The proposed model had optimal performance statistics with RMSE: 0.4 and AMPE: 3.4, although it has poor or large computation overheads with average computation timings at 100.3 s. Thus, the proposed methodology shows fewer RMSE and AMPE errors with improved performance at the cost of increased prediction times. On the other hand, Shallow NN and regression methods seem to have vast prediction errors but fewer computation overheads.

4.4.6 Comparison Between Existing Approaches and The Proposed Model: Computation Speed

This study also evaluated the real-time implementation feasibility and computational performance benchmark of the proposed LSTM prediction architecture. This study used the prediction results on Intel(R) Core (TM) i7-8550U CPU @ 1.80GHz with 16 GB RAM. The computation time results were obtained for all three scenarios' data and compared to the NN model, regression model, and the proposed LSTM method. A comparative analysis of prediction time was carried out between the proposed and other studied architectures. The evaluated prediction times for all of the architectures are presented in Figure 4-20. The analysis also measured the percentage increase in computation times across all three scenarios and covered both temporal and environmental aspects. This analysis is presented in Figure 4-21.

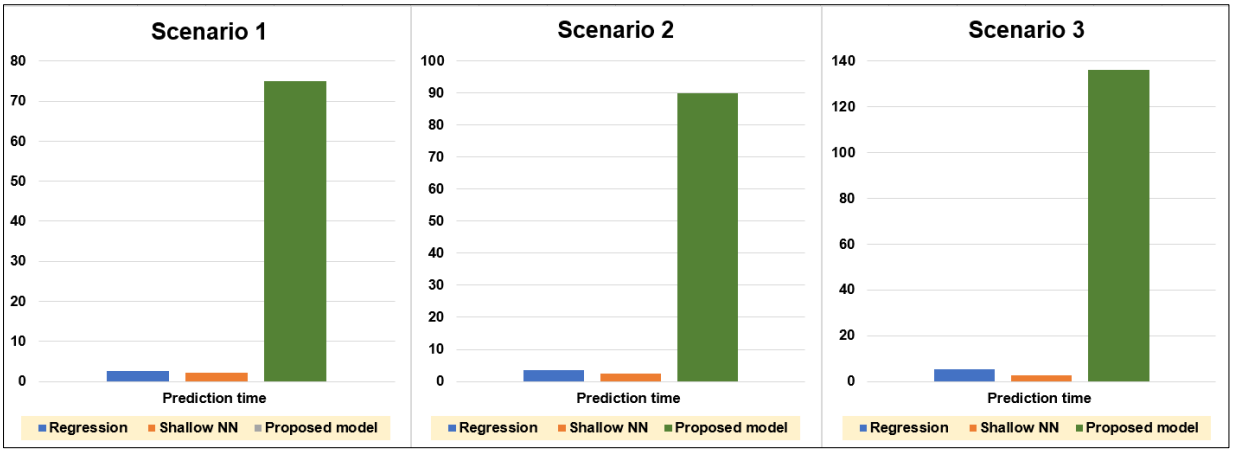

Figure 4-20 Computation timing for shallow NN, regression, and proposed model

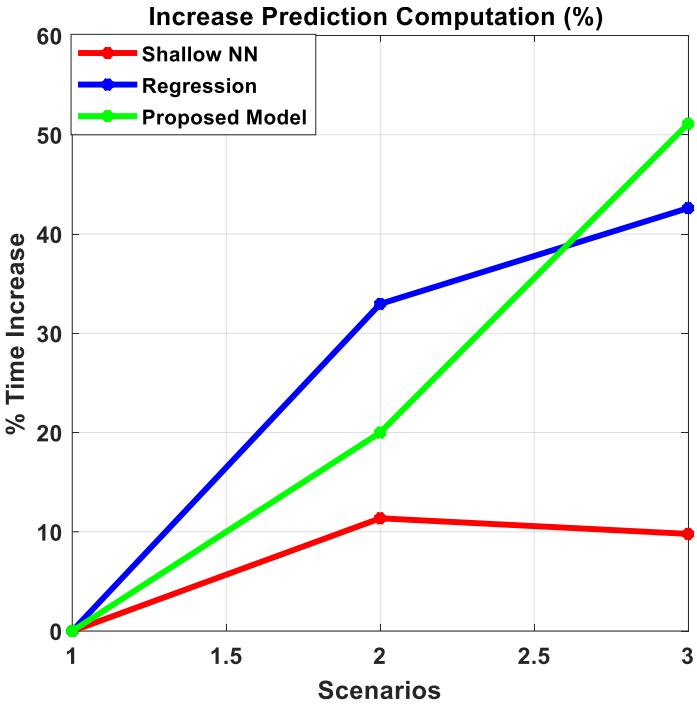

Figure 4-21 Prediction timings percent increase

It is also evident from Figure 4-21 that the prediction times for shallow NN-based prediction steadily increases by 10% to 11%. On the other hand, the increase in prediction times for regression and LSTM methods was even more significant at 32% to 42%, and 20% to 51%, respectively. Thus, it is clear that the percentage increase of prediction times remains steady for shallow NN but increases faster for regression and LSTM-based prediction methods between 9:00 AM to 12:00 PM from scenario 1 to scenario 3.

It is deduced from the above results that the prediction time of the shallow and NARX networks is less than the proposed LSTM architecture. However, prediction times can be further reduced by increasing the computational effort and employing larger computational machines.

4.5 Conclusions and Future Work

There is a clear need for accurate congestion forecasting in UTM systems. This research study proposes a model to predict air traffic congestion using existing machine learning techniques and adapted complexity metrics based on a linear dynamic system. In contrast to previous studies [218] that have assumed a static environment, or have deployed a grid mission network model (as UAS traverses a region along a structured grid) [271], the proposed model was validated using a drone delivery system scenario which in turn addresses uncertainties generated by adverse weather, as well as dynamic and static obstacles. The present study contributes to the limited research on the prediction of air traffic flow for UTM systems, addressing the limitation that the existing literature covers either trajectory prediction [272], [273] or conflict detection and resolution [274], [275]. Deployment of the model aims to address issues related to UAV traffic management (e.g., congestion). In existing works [218], [232], the CNN model has been employed to predict density in an urban area. Nonetheless, one of the major disadvantages of CNNs is the excessive cost of computation [276], [277]. The present study proposes a DL model comprising LSTM and 1D-CNNs. This will have the dual benefit of enhancing deep neural network accuracy and saving time.

Based on complexity prediction graphs reflecting air traffic hotspots, the proposed model permits UTM operators to reconfigure and regulate UAV paths. It may be employed to recommend suitable conflict-avoidance trajectories through the manipulation of ground delay, speed, or heading, while also allowing congestion in predicted UAV-traffic hotspots to be reduced. By forecasting congestion areas in advance, and by facilitating suitable actions to prevent their emergence, the proposed model lightens the workload of air traffic controllers.

Severe weather, airspace restrictions arising from emergency operations, and potential collisions with both dynamic and static objects, all generate uncertainty within the UTM operational environment. In consequence, the traditional air traffic control paradigm makes it impossible for air traffic controllers to ensure the safety and efficiency of operations. It is, therefore, of great importance that a new decision support system (DSS) be designed as soon as possible, and specifically, one attuned to the needs of integrated manned/UAS Traffic Management. It is technically possible to construct highly automated UTM systems that deploy artificial intelligence (AI) algorithms. Nonetheless, intelligent algorithms typically evince opacity, and are deficient in transparency and inexplicability, which restricts their practical application [278]. With these challenges in mind, future work by this team will involve recommendations for a decision support system to aid UTM controllers. The system will help the latter predict the safety responses necessary to mitigate congestion hotspots—which would otherwise jeopardize airspace safety. This advisory system would accommodate factors such as environmental conditions, UAV states, obstacle maps, and mission priorities. In doing so, it would also calculate the likelihood of mission failure before re-planning as required. Thus, to promote sustainable trust and acceptance by end-users, this advisory system would emphasize the ‘explainability’ of the decision support system (DSS) resolution, in terms of safety-critical features and certification [279].

5 ASSURING SAFE AND EFFICIENT OPERATION OF UAV USING EXPLAIMABLE MACHINE LEARNING

5.1 Introduction

Despite the increased popularity of Unmanned Aircraft Systems (UAS) in recent years, the relevant infrastructure remains immature. Nor is any managerial model available to permit the safe and efficient operation of UAS within low-altitude airspace [280]. This has generated a clear demand in terms of a viable operational framework, in response to which European authorities and NASA have, respectively, proposed plans for “U-Space” and USA Traffic Management (UTM) [20]. UTM, nonetheless, must still conform to the rules and regulations associated with Air Traffic Management (ATM), and this requires a flexible approach towards airspace regulations.

Air traffic-flow and capacity management, or ATFCM, has become increasingly important as demand for conventional air transportation has risen. Alongside airspace management and air-traffic services, ATFCM is, in fact, one of the acknowledged “three pillars” of ATM. The primary goal of ATFCM is to facilitate an early-stage safety net to prevent the overloading of air-traffic control (ATC), and it pursues this aim by balancing airspace capacity and traffic demand. Moreover, airspace thresholds will soon be reached due to the rapid increase in *UAS* demand, and this may require the development of similar ATFCM initiatives [281]. A Dynamic Capacity management service should be achieved at the U3 stage, according to the U-space roadmap. Given a higher degree of autonomy, stage three could include capacity-management support for conflict detection, as well as more complex operations in areas of high density. In due course, interactions between manned aircraft and ATM/ATC will be a matter of routine [282]. Indeed, given the steadily rising number of drone operations, most proposed operational UTM concepts, even internationally, acknowledge the difficulty of realizing a continuous DCM process to support such operations. Meanwhile, as compared with the DCM of extant ATM systems, any new process will evince significant differences. This is due, among other factors, to the nature of the drone market, with its greater divergence of aircraft types, business

models, and anticipated technologies for Communication, Navigation and Surveillance (CNS) [283]. UAS/UAM operators, compared with traditional airlines, will present a more heterogeneous array of preferences, driven by the capabilities of their vehicles and the nature of their missions [284]. In terms of delay costs, for example, there may be a wide discrepancy between an aerial platform for monitoring pollution, and a package-delivery assignment. Speed restrictions, holds, and airborne delays may exert a greater impact on fixed-wing drones, or vehicles of limited endurance and range, than on (e.g.) rotary-wing drones. Moreover, it will be difficult to anticipate the use of vertiport resources and airspace in the context of UTM demand, since the latter is likely to be highly dynamic. Such problems are compounded by the fact that UAS operators may submit flight plans with varying, or little, advance notice. On-demand mobility applications of this kind must be supported by future UTM systems. Finally, there will be a pressing need for UTM-mediated congestion management, given the very large scale of expected UAS/UAM operations. The complexity of operations will also exaggerate the effect of any unfair allocations, which may affect thousands of flights per hour.

Within the service structure of ATM [213], the key mission of Demand and Capacity Management in airspace (for the sake of safe, efficient air traffic) is undertaken by Air Traffic-Flow Management (ATFM) [285]. Airspace-management systems and ATFM itself (as based on Artificial Intelligence (AI) algorithms) have proven highly successful in reducing both delays and congestion [286]–[288]. Conversely, high-density traffic in low-altitude environments cannot be accommodated by current ATFM, in terms of either timeframe or intensity. Thus, to address the specific demands of dense, low-altitude, urban airspace, and dynamically to respond (in real-time), to both UAS states and airspace changes *per se,* it is essential to develop an intelligent UTM system that incorporates appropriate DCM technologies and processes.

While much research has been carried out in terms of AI for the ATM field, little or none of this work has brought tangible advantage to end users or, indeed, been fully translated into real-world operations. This slow progress reflects the fact that

safety must be prioritized in fields such as ATM, where human lives are potentially at risk [289]. Thus, AI will be a key component of decision-support systems for future ATM and UTM. This will provide a degree of liberation from the limited flexibility of the algorithmic logic that one usually finds in declarative automation [287]. Ongoing research into human-machine interaction (HMI) is being driven by increases in both processed-information volumes and automation complexity, and this should enhance human-machine coordination [290]. In order to improve the assistance rendered to the decision-making process of the ATM operator, meanwhile, ATM DSS have progressively evolved [291][292]. Conversely, important concerns regarding ethics, privacy and law have arisen, due to an absence of “explainable features” and knowledge representation; such absences undermine the human ability to monitor, or even comprehend, proposed solutions. Since safety-critical systems need to be traceable, this is unacceptable [293]. Consequently, there is a trend away from Black-Box responses to operational challenges in military, business or even personal life. Such concerns have, in turn, driven an appetite for systems that provide transparent, comprehensible attributes for their machines. Ideally, such systems permit human users to understand (i) the AI algorithm itself (global interpretability and explanation), and/or (ii) the solutions it generates (local explanation and justification). The methods of Explainable Artificial Intelligence (XAI) are closely linked to the systems that the latter seeks to explain; indeed, the approach is already in its third generation [294]. The XAI paradigm seeks to generate a suite of machine-learning methodologies that will sustain high prediction accuracy (learning performance) in tandem with greater explicability.

The evolution of complexity metrics presented in Chapter 4 paves the way for the development of a demand and capacity management system (DCM). These metrics enable the distribution of congestion figures in the UTM airspace to be accurately determined. This chapter uses this information as input to form a well-defined methodology for measuring realistic traffic flow and the capacity of the airspace in question. Once the true capacity is known, an explainable decision-making model based on congestion levels and the priority of incoming UAV

missions is leveraged that takes care of both the capacity and safety of the UTM airspace. Specifically, a rule-based decision tree has been formulated using the available capacity of congested zones, new incoming UAVs trajectories, and mission-priority information. Hierarchical in nature, decision trees are structures used to address problems of classification and regression, and the decisions that such areas involve [295]. They have long been a feature of various categories of transparent models, and their understandability and complexity have always been viewed as important variables, given that they have been associated with so many decision-making contexts. In fact, off-the-shelf transparency is regarded as a key advantage of these structures. Their heterogeneous applicability, moreover, means that experts from one field are frequently content to accept the results of decision trees formulated by specialists in another [296]. The overall contributions of this chapter can be summarized below:

1. This work introduces a hybrid, explainable machine-learning model that enhances UTM demand and capacity-management services through the estimation of airspace capacity. The proposed model utilizes a traffic flow analysis framework and an LSTM-based congestion prediction model, which were presented in chapters 3 and 4, respectively. The rule-based module recommends efficient and safe UAV mission routes based on measured airspace capacities.
2. This thesis proposes, in addition, a data-analytics framework to characterize traffic flow patterns for UTM airspace, essentially by analyzing simulated historical data. The methodology focuses on two main components that intervene in a DCM process, namely, the prediction of congestion figures for each trajectory, and the accurate estimation of airspace capacity. Specifically, we identified five congestion levels, and a clustering algorithm-based mechanism was developed to determine available urban airspace for Urban Air Mobility (UAM) operations, as based on the UTM traffic-flow analysis.
3. In terms of the explainability of the decision-support system, this study proposes a transparency-based methodology with a fusion of both Black-Box and explainable White-Box models for our UTM recommendation systems. The Black-Box models are not transparent, due to a lack of clarity associated

with their internal configuration. By contrast, White-Box models manifest observable and understandable behaviors. We have introduced metrics-based scoring to illustrate the overall explainability of our hybrid model, based on the transparency of the individual components. In light of these metrics, we have confirmed that our proposed advisory system is approximately 70% explainable.

The structure of the remainder of this chapter is as follows: Section 5.2 presents a review of the Explainable Artificial Intelligence (XAI), analyzing why XAI is needed, how it is currently provided, and its limitations. Section 5.3 describes the proposed methodology of this project. Section 5.4 presents the explainable demand capacity-management system and its results. Finally, Section 5.5 provides the conclusions, together with some guidelines for future work.

5.2 Literature Review and Background

Traditional air traffic-management approaches have been challenged by the developing interest in unrestricted UTM and UAM. ATM systems, if they are to overcome extant and emerging problems, must increase both capacity and efficiency: this, in turn, means high levels of intelligence and automation. The reviewed literature [30], [31], [303]–[309], [32], [33], [297]–[302] , indicates the increasing significance of utilizing AI in Air Traffic Management (ATM) and its continuous development. For instance, the volume of publications pertinent to this subject has increased by 100% in the last four years alone. As AI-ATM technologies mature, furthermore, the need for explicability increases proportionately. If end-users (ATM operators) cannot understand a given system, after all, they are less likely to accept it. The review also demonstrates that data analytics have been applied to virtually all of the more difficult stages of the ATM domain. Despite this, scholarly coverage in the context of UTM has been meager. In seeking to achieve quasi-human, or even superhuman, performance *vis-à-vis* the automation of certain tasks, machine learning (ML) has become the most favored tool. In terms of unmanned aircraft, moreover, the usage potential of ML is especially high. Emerging concepts of Urban Air Mobility envisage the removal

of the onboard pilot, while a remote pilot (perhaps supervising a fleet of vehicles) will be supported by ML [310]. In fact, UTM operations comprise a vital component of ATM. This is true, in particular, for those beyond visual line of sight (BVLOS), whereby operators may be unable to distinguish between manned and unmanned vehicles, and high volumes of decision making are needed. This reflects the need for a range of services, including remote identification transmission, alerts for in-flight conflicts, weather forecasting, location of nearby UAS operators for data exchange, strategic deconfliction via negotiation and/or flight-intention sharing, and deconfliction maneuvers [15]. In this respect, ML instruments and techniques are a promising solution, since they can deploy predictions as a route towards superior decision-support systems. Today, indeed, ML-based decision-support systems can already identify patterns of interaction between variables, thereby resolving problems of complex mass-data analysis [311].

The principal problem associated with machine-learning models, nonetheless, is the perception that they are Black-Box solutions. In other words, there is an absence of lucid declarative knowledge representation, even if one broadly understands the mathematical principles that underpin them [312]. With the majority of ML models, furthermore, neither the output results nor the algorithm in question are explainable: consequently, operators who cannot understand solutions may be reluctant to accept them [313]. Problems concerning certification for ML-based, safety-critical systems will be addressed in the following sections, which will also consider the various solutions that the literature comprises.

5.2.1 The Challenges of Certifying AI

Software applications, reliant on ML, are deployed in many important areas of contemporary life, from finance to health, from energy to logistics. Given its increasing real-world significance, then, the safety aspects of machine learning have become the focus of increasing attention [310], [314], [315]. Indeed, especially in the case of safety-critical applications, the incorporation of ML is not without risk. In such circumstances, any serious breakdown can have disastrous

consequences, up to and including the loss of human life. In the automotive and avionic sectors, additional damage may be sustained by the environment, or by expensive equipment.

In the context of ML systems, failures are interpreted as harmful or unintended behavior, and this may arise if ML is poorly incorporated into the system in question [316]. ML algorithms are probabilistic in character, but this may not sit well with safety cultures that exist, or emerge, when safety-critical systems are developed [317]. In contrast to conventional software, the nature of such algorithms is not widely understood. ML algorithms may evince a comparatively clean and static structure, but in order to function, they require numerical parameters extracted from various datasets [318]. Unlike traditional software, there is a lack of explicit programming oriented towards particular tasks [319]. Before any ML-based components can be utilized within a safety-critical system, these components must be properly certified.

One extant study [320] defines certification as a “procedure by which a third-party gives written assurance that a product, process, or service conforms to specified requirements.” In avionic or automotive contexts, where safety is a critical issue, certification is obviously important. A range of standards have thus been developed to address this requirement. For example, IEC 615803 provides an international standard for the certification of safety-related electrical, electronic or programmable electronic items. The same standard, enhanced for purposes of road-vehicle safety, is represented by ISO 262264. Meanwhile, DO-178C5 [321] has been developed to address the certification of airborne equipment and systems. This standard, for example, introduces the requirement for modified conditions/decision coverage (MC/DC) criteria, the objective being to interrogate all the potential conditions that might give rise to a particular decision. These standards and others like them, in sum, are specifically designed to accommodate functional considerations of safety within safety-critical systems.

The significance of the topic is reflected in the recent emergence of several certification initiatives, such as the French Dependable and Explainable Learning (DEEL) Certification Workgroup, as well as working groups on standardization, such as EUROCAE WG-114 and SAE G-34 [322].

A large number of surveys reflect the burgeoning body of research dealing with ML. The great majority of these, however, do not explicitly consider assurance, but rather focus on specific types of ML. One finds, for instance, surveys about deep learning [323], [324], reinforcement learning [325], [326], transfer learning [327], [328], and ensemble learning [329], [330]. Certainly, in terms of the forms of ML that they address, these surveys offer valuable insights into the efficacy, trade-offs and applicability of the available data-management and model-learning methods. These surveys do not consider assurance-related techniques for the data-management and model-learning phases of the ML lifecycle, however, and (to repeat) they prioritize specific types of ML and specific classes of verification technique. Moreover, the safety aspects of ML-component integration within autonomous systems are addressed briefly, if at all.

Still, a few recently published surveys do provide comprehensive information regarding state-of-the-art ML assurance, namely, the evidence generated to identify whether ML is safe enough for its envisaged purpose. Such surveys [315], [322], [331], [332] consider the various ways in which this evidence may be created at particular points during the ML lifecycle. The latter is an iterative, complex process that begins with the harvesting of data to be used to train an ML component of a given system and finishes with the real-world deployment of that component. The studies cited above initially provide a systematic description of the various stages of the ML lifecycle. Then, for each stage, they specify the relevant assurance desiderata.

In the aviation sector, AI is receiving much attention, and it is already producing tangible results. Before AI/ML can be deployed more widely in ATM operations, however, important challenges must be overcome. There is as yet, for example, no commonly accepted framework for ML to be employed in the sphere of aviation [310], [333]. First guidance for the safe usage of ML in this regard has been proposed by the European Union Aviation Safety Agency (EASA). The EASA AI Roadmap was published in 2020 [334], and this provides basic definitions and term clarifications for the future aviation employment of ML technologies. It also outlines some of the problems that must be addressed.

Indeed, EASA examines the probable influence of ML in a variety of areas, including the design, production, operation and maintenance of aircraft, as well as air-traffic management.

Assurance is reinforced by ML interpretability [331], as the latter offers evidence for the following: (1) justifying results (that is, explaining decisions for particular outcomes, especially in the context of unexpected decisions, as well as justifications required for legislative compliance); (2) malfunction prevention, and the identification and correction of errors (i.e., an understanding of system behavior affords better visibility regarding unknown shortcomings and weaknesses); (3) assistance of model improvement (that is, the more easily a model can be explained and understood, the more easily it can be enhanced); and (4) support for understanding the operational domain (i.e., a helpful tool is provided for learning, information harvesting, and – thus - knowledge acquisition) [335]. One may find millions, or billions, of parameters in ML and DL models, with the most successful structures being highly complex and challenging to explain [336]. Interpretability does not guarantee safety in itself, but it can at least shed light on how, and why, models may fail.

5.2.2 State-of-the-Art ML: Methods of Interpretability

The autonomous operation of unmanned aircraft systems (UAS) involves safety-critical issues, and one must know how and why decisions are arrived at. Thus, much research has been conducted on ML architectures, with a view to making ML systems more transparent. EASA, indeed, regards “explainability” as one of the cornerstones of trustworthy AI [334]. The following provides a summary of explainable-AI terms and concepts. Subsequently, examples from the literature will support a more detailed discussion of topics such as post-hoc explainability, explainability metrics, and transparent models.

5.2.2.1 Explainable AI

AI stakeholders are demanding greater transparency, as Black-Box machine-learning (ML) models have become increasingly common in the context of key predictions for critical contexts [337]. Decisions that appear illegitimate or

unjustifiable, or simply defy attempts to explain underlying behavior, are problematic in numerous ways [338]. In areas such as precision medicine, where models must supply experts with far more than simple binary predictions, model outputs must be supported by intelligible explanations [339]. Similar areas, in this regard, are security, finance, and the use of autonomous vehicles. There is a natural human reluctance to rely on methods that are not traceable or readily interpretable, particularly in safety-critical applications where reproducing scenarios of incidents, are fundamental to the investigation at hand [340]. There tends to be a trade-off between model performance and transparency, which has fueled assumptions that prioritization of performance generates opacity [341].

Arrieta et al. [342] define explainable AI as follows: “Given an audience, an explainable AI is one that produces details or reasons to make its functioning clear or easy to understand.” EASA, meanwhile, defines explainability as, “Capability to provide the human with understandable and relevant information on how an AI/ML application is coming to its results” [334]. The definitions are similar, but that of [342] is more exact, since it reminds us that explainability also depends on the relevant target audience. The key characteristics of an explainable model have been outlined as follows [342]:

- **Trustworthiness**: An ML model cannot realistically be deployed without a basis of trust. Otherwise, users may simply ignore model output. As noted above, EASA thus regards explicability as central to trustworthiness, and the latter is one of the key objectives of their AI roadmap [334].
- **Causality**: An additional objective of “explainability” is to facilitate the finding of causation between data variables. For models that assess UAS systemic health, for instance, explainability may reveal that a given component tends to fail after a certain load time.
- **Transferability**: Explainability can also help to clarify model constraints and limitations. Models learn to solve particular problems during training, but an understanding of boundaries is required to ascertain how, or if, the model may be applied to other problems. If a model has been trained to detect obstacles in daylight, for instance, it should not be used at night, at least without suitable modification.

- **Accessibility**: Explainable models will reassure non-expert users, who may feel intimidated by algorithms that, at first glance, appear inexplicable.

In order to furnish future developers with design options to address the performance/explicability trade-off, as cited earlier, various techniques have been considered. Recent progress towards the construction of viable XAI systems has been encouraging. As shown in Figure 5-1, the literature has identified at least two key elements of explainability. The first pertains to ML systems that have been specifically designed with human interpretability in mind: these are so-called "transparent ML models." The second pertains to the explanation of models that are actually shallow; this is termed "post-hoc explainability." Based on the above characteristics of an explainable model, the following sections consider both these aspects in greater detail.

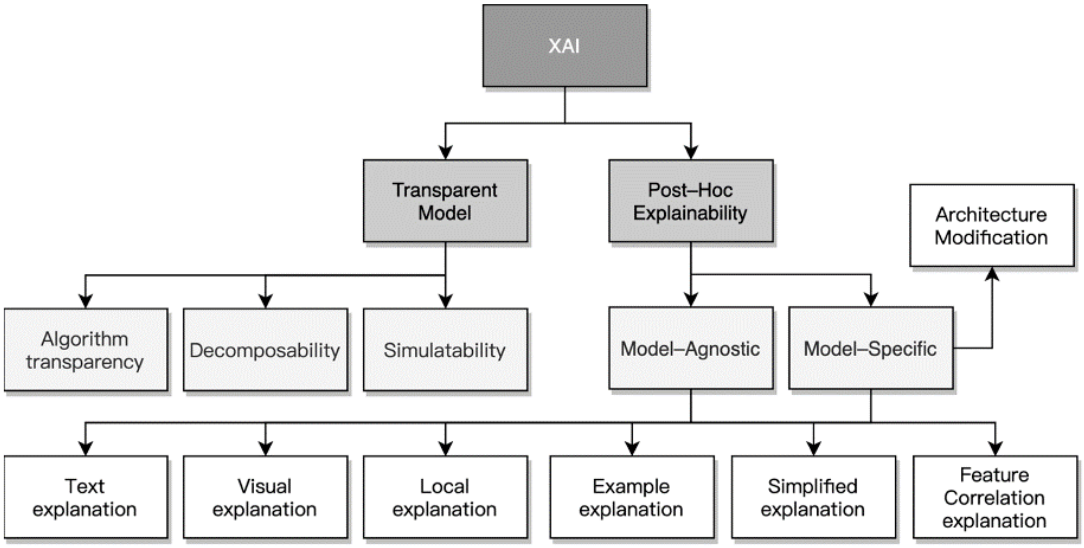

Figure 5-1 Classification of explainable AI methods [279]

5.2.2.2 Transparent Models

Models are considered transparent if they are intrinsically and independently understandable. They are unlike Black-Box systems, since their architecture affords numerous insights regarding their internal relations [310]. Still, transparent models can afford varying degrees of explainability, and to distinguish between these, three categories have been postulated. These are (1)

simulatable, (2) decomposable, and (3) algorithmically transparent models [343]. Simulatability represents the highest degree of transparency. The next degree is that of decomposability, followed, finally, by algorithmic transparency.

More specifically, simulatability references system interpretability, whereby human beings can simultaneously understand and simulate algorithmic behavior. Conversely, highly complex systems, incorporating numerous features, are challenging for humans to grasp, and perhaps even impossible; this remains true even with the deployment of inherently transparent algorithms [344]. Decomposability per se indicates that parameters, input, calculation - in fact, each part of a given model - should accommodate an intuitive interpretation [342]. At the level of algorithmic transparency and the learning algorithm itself, a final notion of transparency might be applied. In other words, the method of model training would itself be understandable [345].

Logistic regression (LR) is a form of classification model. This model assumes linear dependence between predicted variables and predictors. The “stiffness” of the model is the particular factor that allows it to be deemed a transparent method [342]. Additionally, decision trees easily satisfy the criteria for transparency. Used to support regression and classification challenges, they are essentially hierarchical decision-making structures [346]. In their simplest form, decision trees are also simulatable models. Depending on their properties, moreover, they can also be algorithmically transparent or decomposable.

When a model generates rules to characterize the data it will learn from, meanwhile, this is termed “rule-based learning”. Rules may take a simple form, such as if/when conditions, or knowledge may be formed from more complex combinations of rules. Rule-based learners, indeed, are clearly transparent models; since they generate rules to explain their predictions, they are often deployed to explain complex models [347], [348].

One issue with transparent models, however, is the generally assumed trade-off between explicability and performance: i.e., compared with more complex models, explainable or transparent counterparts are thought to be less accurate [338], [342]. There is no hard scientific evidence to support this belief, except for

some applications in which transparent models might, for instance, be outperformed by opaque DNNs [349]. In [349], furthermore, one finds cases in which the performance of a transparent model is equal, or even superior, to that of a less transparent structure. Neural circuit policies (NCPs) are an instance of model architecture that performs better than comparable, less transparent models [350]. The model described in [350] performs as well as a long short-term memory (LSTM) RNN, with identical convolutional layers. This is true, despite the model having less than one twentieth of the trainable RNN parameters, and only 19 RNN neurons.

5.2.2.3 Post-hoc Explainability

The [342] presents various approaches to post-hoc explainability, including local explanations, feature relevance, explanations by example, visualization, model simplification, and text explanations. Among these approaches, local interpretable model-agnostic techniques (LIMEs) serve as examples of model-agnostic techniques, as implied by their name [351]. These provide local explanations, or more exactly, explanations for single outputs and their local environments, and LIMEs generate linear models to approximate the original models for those environments. These linear models can then be deployed as explanations for the output. Shapley additive explanations (SHAPs) are further examples of model-agnostic technique [352]. SHAP offers an alternative explanatory method for local ML. Based on the game-theory principle, it evaluates the significance of additive features for each particular prediction [353]. A tree explainer based on SHAP has been developed by Lundberg et al. [354], thus providing a visual method of local explanation. The local explanation is extended, while feature values and weights are assigned, in order directly to capture the interaction of features. The global structure is comprehended via a large number of local explanations.

For computer-vision-related tasks, Convolutional Neural Network (CNN) are highly favored. As with other opaque DNNs, however, their predictions are extremely challenging to explain. Consequently, numerous post-hoc techniques have evolved, specifically to clarify CNNs [355]–[357]. The majority of these approaches employ saliency maps, the latter comprising a mixture of feature-

relevance explanation and visualization, whereby the impact of each pixel on the prediction is computed and visualized. In addition to the methods referenced above, it is also possible to use text explanations to render model behavior more comprehensible. A Deep Neural Network (DNN) for natural language processing (NLP) is employed, in [358], to predict beer-review ratings, and to retrieve questions from a web-based forum. The DNN emphasizes short, coherent elements of the text that are sufficient to explain the prediction. Yet another means of explaining the decisions of models is provided by “meaningful examples” also known as “explanations by example”. The method of testing with concept-activation vectors (TCAV), meanwhile, is presented in [359]. This approach permits the learning of various high-level concepts, such as “horse,” “stripe” or “desert”. The relevance of such concepts is computed when classification tasks are required, such as “detecting zebras”. The decisions and predictions generated by machines must be explainable, or their reliability will be questioned. A model of breast-cancer diagnosis, via XAI visualization methods, for example, was presented by Lamy et al. [360]. In parallel, clinicians were provided with a graphical user interface, both for the sake of usability and to reinforce “acceptability.” Medical staff benefited from these provisions, since they needed, not merely to be aware of recommendations, but to be confident in their suitability and efficacy.

Widely recognized metrics are necessary if one needs to assess, or compare, different approaches to explainability. Some metrics-based studies have considered the problem of measuring the ML-model explicability, or post-hoc methods [361], [362]. Most metrics of this kind (utility, goodness, trust, satisfaction, etc.) are difficult to quantify, however. In most cases, a human evaluation of explainability is required, as the authors of [363] point out. Metrics, either singly or in groups, should facilitate a meaningful appraisal of how closely models conform to the definition of explicability. Classic metrics (F1, accuracy, sensitivity, etc.) illustrate the efficacy of model performance in terms of a particular aspect of explainability. Recent attempts have been made to improve the measurement of XAI, as detailed in [361], [362]. Generally, measurements of XAI should assess the impact of model explanations regarding usefulness,

“goodness”, satisfaction, and “improvement of the audience’s mental model.” They should also evaluate the effect of explanations on model performance, and on the audience’s reliance and trust vis-à-vis the model.

5.3 Proposed Advisory System Framework

5.3.1 Overall Framework

The establishment of a UTM system is necessary, given the increasing quantity of UAVs. A key challenge here is capacity estimation. In other words, within a particular airspace, how much traffic can be safely and effectively managed? There are various perspectives through which this question can be approached. One must also take into account a number of factors that limit capacity, such as: excessive noise (better technology may be required to improve public perception of the drone industry); the emergence of hard-to-resolve conflicts (measures for capacity management must be deployed if their likelihood becomes too high); and jamming of the communication spectrum (since stronger encryption protocols demand greater bandwidth, this might include cybersecurity factors) - etc. In sum, given a set of operational requirements, safety, stability and noise conditions, protocols and technological capabilities, how many aircraft can the airspace in question accommodate?

Methodologically, in order to address the performance/explainability trade-off, the present research deploys a hybrid model with a combination of Deep Learning (for accurate prediction of congestion), iterative DBSCAN (to identify physically congested sub-regions), and rule-based decision logic (for better explainability of UTM management decisions). This gives rise to an intelligent system that will be reasonably comprehensive and explainable. The overall architecture of the proposed model is presented in Figure 5-2. The detailed steps of the proposed methodology, meanwhile, are provided below:

1. The process of UAV trajectory-data generation is performed using PSO simulation. It is noteworthy that the dataset employed in this process is derived from a previously generated dataset, as discussed in the preceding chapter.

2. The pre-processing of acquired data is expanded by up-sampling, in order to increase the resolution of UAV trajectories. This generates better air traffic-flow and congestion analysis.
3. An LSTM-based congestion-prediction model, as described in the previous chapter, has been applied to generate predicted congested values for individual trajectories. The predicted congestion values are normalized between 0-100%, in order to threshold the congestion levels.
4. In this work, the quantitative definition of congestion is derived from a complexity metric (Equation 4-6) that specifically measures the potential complexity of future trajectories near each UAV. The details of this metric are presented in Section 4.3.3. However, it is important to note that the metric is normalized to represent congestion as a percentage, ranging from 0% to 100%. Additionally, to illustrate the distribution of congestion levels within the Bedfordshire airspace (64 km x 64 km), the congestion percentages are further categorized into five classes: 0-20%, 20-40%, 40-60%, 60-80%, and 80-100%. In Table 5-1, five congestion levels have been defined for the purpose of facilitating subsequent analysis:

Table 5-1 Congestion levels

Level No.	Congestion Range%	Congestion definition
Level-1	00%-----20%	Lowest
Level-2	20%-----40%	Lower
Level-3	40%-----60%	Medium
Level-4	60%-----80%	Higher
Level-5	80%-----100%	Highest

5. Considering the distribution of congested levels across the entire Bedfordshire UTM airspace (64 km x 64 km), the congested zones or sub-regions corresponding to each of the five congestion levels have been identified. This can be done by running and tuning the DBSCAN clustering algorithm, iteratively, for each congestion level. The optimal tuning is done by adjusting the parameters “eps” and minimum points (“minPts”) for

DBSCAN. The parameter tuning is required for better trajectory cluster-grouping formation; moreover, it also helps in defining better congestion-area polygons.

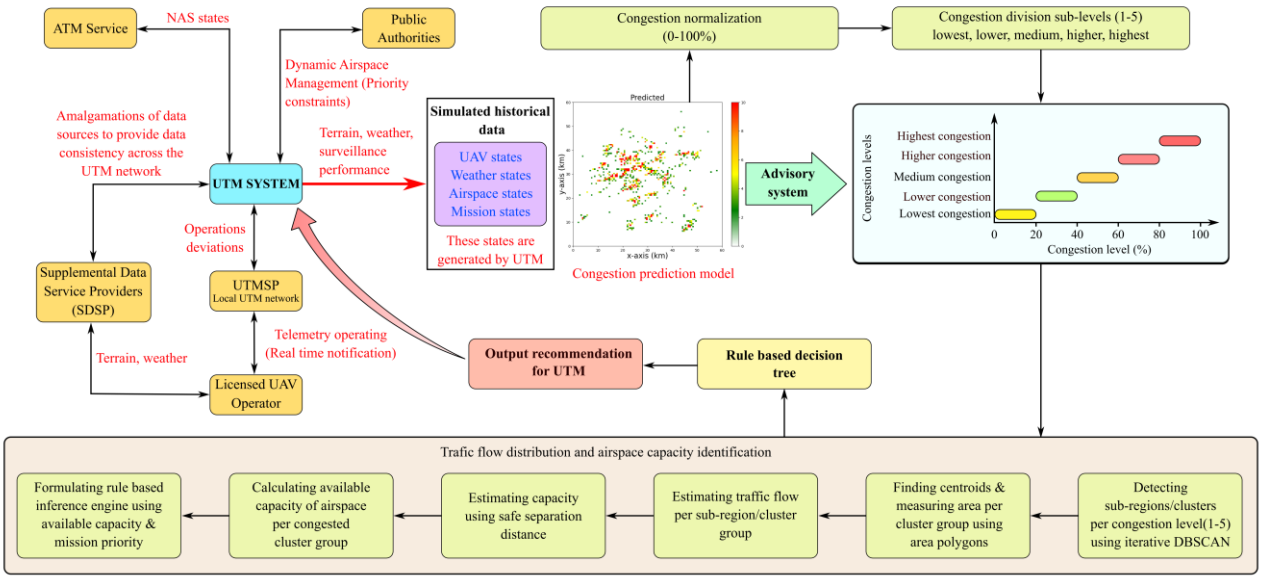

Figure 5-2 The explainable machine-learning model for the safe and efficient operation of UAV

6. The area polygons are created around these congested clusters or groups, both to estimate the covered area per cluster, and to locate the centroid position (x, y) around which a cluster is formed. The covered area around these clusters is built by forming an irregular polygon (using the boundary points), and by measuring the area using the MATLAB poly-shape function. The count of UAV trajectory points for these congested zones is also measured.
7. The traffic flow for each congestion cluster is calculated using the ratio between UAV trajectory counts and the area encapsulated by that cluster.
8. This study focuses on a spatial 2D (x, y) capacity measurement analysis, assuming a uniform altitude level for all UAVs within the airspace. The capacity of the airspace is determined based on the concept of safe separation distance. Specifically, a lateral separation distance of 100m is

considered, while the vertical distance is not taken into account in this work. This, in turn, indicates 10 UAVs per km, which implies about 100 UAVs per km2 within each cluster. The available capacity for each cluster is calculated by taking the difference between the current traffic flow and the traffic-flow threshold (100 UAV trajectories/ km2).

9. Currently, there is no established standard for timing constraints in measuring UTM airspace capacity. However, the ATM domain offers suggestions, such as defining sector capacity in hourly intervals [364][365]. In line with this, the study conducts capacity calculations on an hourly interval, taking into account the average endurance of UAVs (20 minutes) and the random start times of UAV missions.
10. The rule-based decision tree is then designed and implemented for each of the five congestion levels (lowest-highest) using three inputs: 1) available airspace capacity (capacity per cluster), which is computed via our congestion analysis; 2) the number of new incoming UAV trajectory points that happen to traverse the congested regions (lowest to highest); and finally, 3) the mission priorities of incoming UAVs, which are required for optimal recommendations. The output of the advisory system is the updated capacity, either allowing the UAV mission within a particular congestion cluster, or disallowing (for safety reasons) the usage of a particular congested airspace. This is followed by a recommendation to use specific, available airspace.

5.4 Results and Discussion

5.4.1 UTM Congested Subzones: Identification and Area Distribution

This section introduces a data-analytics framework designed to characterize traffic-flow patterns within the Unmanned Traffic Management (UTM) airspace. Subsequently, the findings from this analysis are utilized to propose an Explainable Artificial Intelligence (XAI) solution specifically tailored to meet the requirements of Demand and Capacity Management services within the UTM airspace.

5.4.1.1 Congestion-Level Identification Using DBSCAN

This section presents the congestion sub-graphs, together with a discussion of the spatial distribution of the congestion hot spots for five threshold levels of congestion (lowest, lower, medium, higher, and highest). The clusters detected per congestion level, as described in this section, help measure the congested areas and their distribution for the entire airspace. The DBSCAN clustering was applied to detect subregions within each level of congestion, throughout the Bedfordshire airspace, utilizing a 64 x 64 km^2 Area. DBSCAN clustering depends upon the input parameters epsilon and minPts [197], and the selection of these parameters was performed utilizing a rule of thumb [366] that, in turn, depends on the number of dimensions (D) in the data set, normally the minPts >= D + 1. For datasets of larger scale, characterized by significant levels of noise, it is recommended to employ a minimum number of points (minPts) equal to twice the dimensionality (D) of the dataset. In the case of a two-dimensional (2D) dataset, a value of 2D+1, which equals 5, is proposed for assignment. The parameter epsilon ($\boldsymbol{\varepsilon}$) was selected to accommodate both the domain knowledge and the current purpose: this purpose was to detect a cluster, in order to encapsulate an area polygon for traffic-flow measurement. Since the lowest-order and lower-order congestion regions were well scattered, one would anticipate a greater presence of clusters and area polygons in these areas , as compared to the higher- and highest-order congested zones, where fewer clusters were expected, alongside more densely populated trajectory points. Thus, epsilon was tuned heuristically, taking the above goal into account. Further details regarding the flight-trajectory data-analytic framework, using DBSCAN, can be found in our previous work [367]. For the sake of clarity, the parameters used to tune the DBSCAN clustering for Scenario 3, and the results for this scenario, are shown below, in Table 5-2.

Table 5-2 DBSCAN parameters per congestion level

Scenario	Congestion Levels	DBSCAN Parameters		Number of Congestion Clusters Detected
		eps	minPts	
	Lowest	0.20	5	20

3	Lower	0.20	5	24
	Medium	0.20	5	19
	Higher	0.12	5	4
	Highest	0.40	5	3

The area encapsulated within each cluster is calculated by forming an irregular polygon around the boundary of the congestion clusters, and by measuring the polygon area using the algorithm outlined Figure 5-2. The congestion clusters, along with area polygons for the congestion levels of Scenario 3, are shown in Figure 5-3. The red dots in Figure 5-3 represent actual trajectory points in each congested cluster region. The yellow dots, conversely, show the centroid locations for each congested cluster region. The black lines represent the boundary around each congestion cluster, used for measuring effective polygon area. The circles or ellipses, encapsulating the congestion cluster, present the maximum radius circle, with centroids as centers.

It may be observed, from the percentage cluster-count analysis (Table 5-2), that almost 62% of the clusters are formed in congested zones with less than 60% congestion levels, while 27% of the clusters belong to regions with congestion levels between 60% and 70%, and the remaining 11% are subject to higher- or highest-level congestion, between 70% to 100%. It can thus be inferred that congestion clusters are widely spread across the airspace for lowest- and lower-level congested areas, with much larger cluster counts. The medium-level clusters appear denser and less widely spread, as compared to those of the lowest- and lower-level congested regions. The sub-regions with higher and highest levels of congestion evince denser grouping, and more centralized regions, when compared to previous congestion-level groups. This observation remains true for each scenario, 1-3. Nonetheless, the spatial locations of the cluster regions continually change, due to the opening or closing of dynamic airfields, adverse rain and wind, and extreme weather fronts.

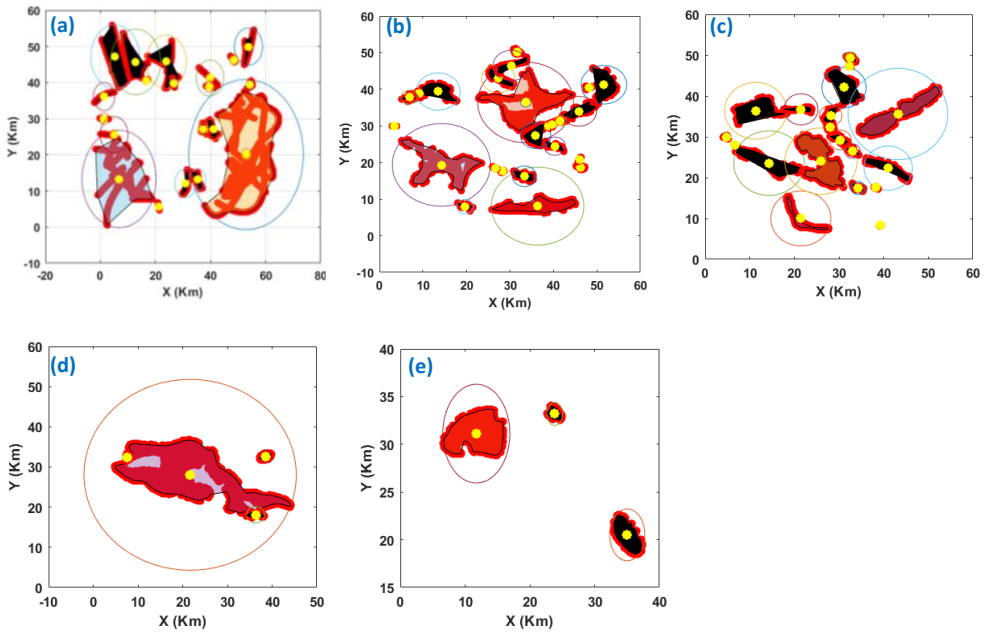

Figure 5-3 Congestion-cluster levels for Scenario 3: (a) lowest level, (b) lower level, (c) medium level, (d) higher level, and (e) highest level

5.4.1.2 Airspace-Congestion Distribution

This section explores the distribution of the airspace area utilized by UAV trajectories in different congested levels, as well as area distribution within each of the congested-level sub-regions in our three scenarios. This area is calculated using the area polygons identified in the previous section, for each cluster and within different levels. The following section provides a comprehensive statistical analysis of the distribution of areas, along with the calculated cumulative area distribution for each congestion level. The data trends clearly demonstrate a more widely spread, and larger, number of clusters for lower congestion regions, as compared to higher- and highest-congestion regions. The congestion-distribution analysis is presented exclusively for Scenario 3, with additional discussion on the tabular results pertaining to Scenarios 1 and 2. The distribution of congested areas per cluster, and the cumulative area withheld by each congestion level under extreme weather conditions (Scenario 3), are shown in Figure 5-4 and Figure 5-5, respectively.

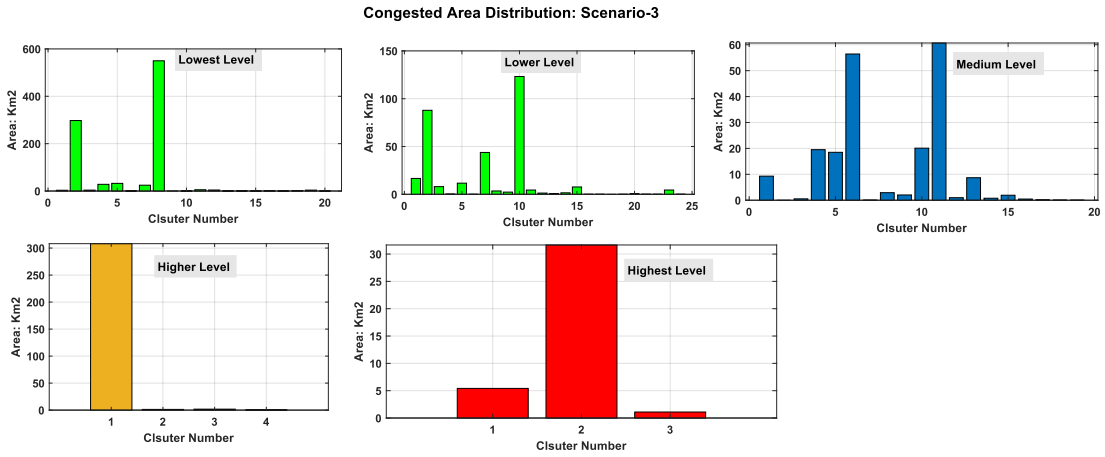

Figure 5-4 Area distribution per level, per cluster: Scenario 3

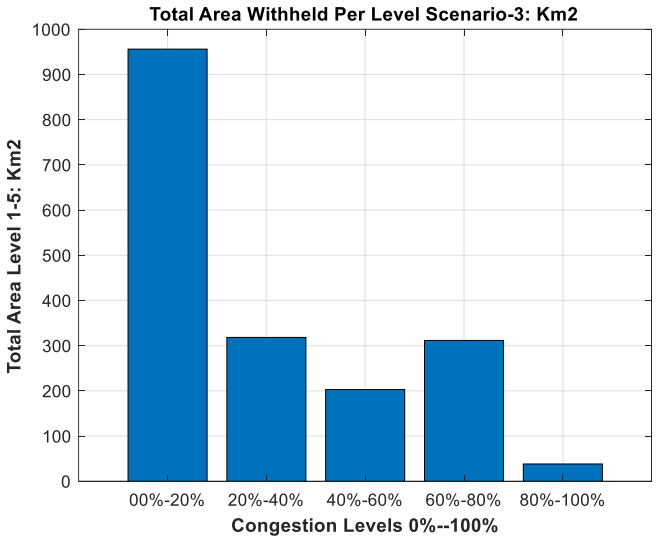

Figure 5-5 Cumulative area distribution per level: Scenario 3

The results in Figure 5-4 indicate that the sub-regions with the lowest and lower levels of congestion evince the highest area usage by UAV trajectory. This utilization of area decreases in the case of medium-congested regions, and it is lowest for highly congested blocks. The cumulative area coverage for each congested level, from lowest to highest, is shown in Figure 5-5. It may be seen, from this figure, that lower- and higher-congested regions encompass almost comparable areas in the airspace. The statistical figures for congested-area utilization in each scenario, 1-3, are shown in Table 5-3.

Table 5-3 Statistics for area utilization ($km^{2)}$ per congestion level, for all scenarios

Scenario 1			
Congestion level	Maximum area ($km^{2)}$	Mean Area ($km^{2)}$	STD ($km^{2)}$
Lowest	264.1664	17.8119	51.7698
Lower	248.8680	23.3337	63.8795
Medium	172.3195	12.3405	42.7612
Higher	135.0970	20.9247	50.4984
Highest	13.3878	7.3998	5.2072
Scenario 2			
Lowest	375.5019	22.0865	69.6584
Lower	122.3201	13.4467	31.0048
Medium	159.9153	25.2928	50.5788
Higher	47.0391	10.6722	16.4358
Highest	80.3852	42.2738	53.8976
Scenario 3			
Lowest	549.8083	47.8039	135.2778
Lower	123.2273	13.2575	30.3633
Medium	60.6678	10.6818	18.2609
Higher	308.1550	10.6722	16.4358
Highest	31.6595	12.7255	16.5392

It is evident from Table 5-3 that the maximum-area peaks lie in the lower and lowest congested regions, and these become smaller as the congestion rises, with the smallest peak lying in the higher and highest congested regions, in *all* scenarios. Furthermore, the standard deviations in the case of no weather constraints (Scenario 1) are relatively consistent for other congestion regions, except the highest congestion area, which experiences smaller congestion clusters. In the case of adverse wind and rain (Scenario 2), the standard deviation is largest for lower congestion regions. The highest congestion area exhibits a much bigger deviation from the mean, due to a large, and much denser cluster of congestion. Nevertheless, under extreme weather fronts (Scenario 3), the

standard deviation is greatest in the lowest-congestion regions. The maximum-area peaks decrease in magnitude as the congestion rises. The mean-area values, meanwhile, are small and consistent. The largest maximum-area peak lies in the higher congested regions.

Figure 5-6 presents the overall percentage distribution of the area being utilized by the UTM airspace environment, the latter including static NFZ, dynamic recreational areas and airfields, weather fronts due to rain, wind and extreme weather conditions, UAV trajectory-congestion clusters, and free airspace. It may be inferred from Figure 5-6 that UTM-free airspace declines from 42% to 30% in the second hour, and further falls to 23% in the third hour: this provides a clear indication of the impact of adverse weather conditions such as wind, rain and extreme weather fronts. It can also be seen that there is an increase in the percentage of UTM-congested airspace, due to UAV trajectories, from 34% to 42% in the second hour (10:00 am to 11:00 am). This further increases to 49% in the third hour (11:00 am to 12:00 pm). The cumulative areas for the lowest and highest congested clusters, for all three scenarios, are depicted in Table 5-4, and this supports the argument above. The congestion-area calculations cited previously were used to compute the traffic-flow capacity in all three scenarios, as discussed in the next section.

Table 5-4 Lowest- and highest-congestion clusters: cumulative areas (km^2)

Congestion Level	Cumulative Area		
	Scenario 1	Scenario 2	Scenario 3
Highest	498.7	750.9	956.1
Lowest	22.2	84.55	38.1

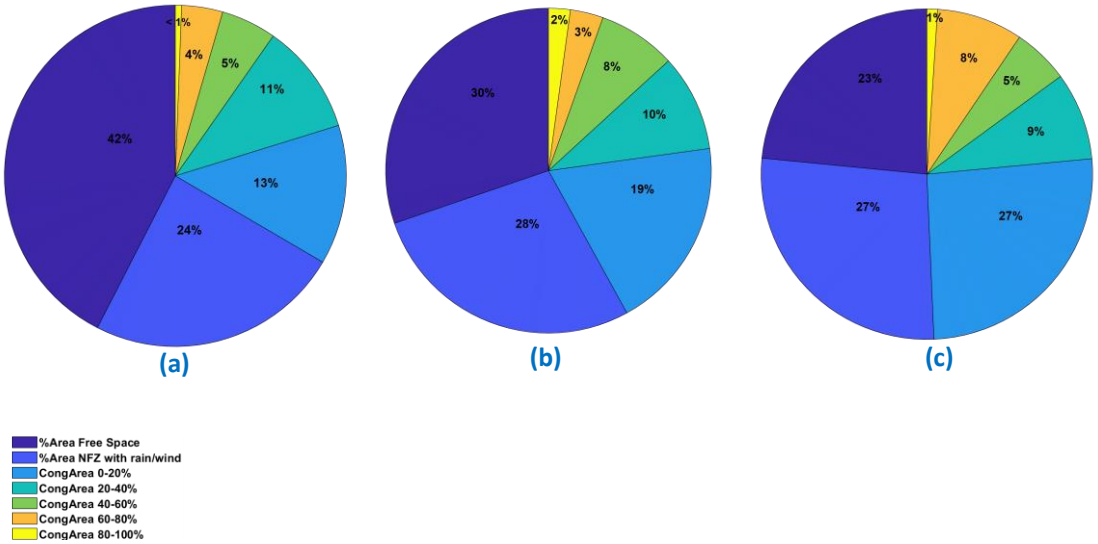

Figure 5-6 Percentage congestion distribution for all three scenarios: a) first, (b) second, and (c) third scenario

5.4.1.3 Traffic-Flow Distribution and Airspace-Capacity Identification

This section addresses the traffic-flow distribution and capacity management for each of the three scenarios. More specifically, the traffic-flow distribution per cluster and per congestion level is presented below. Assuming a safe separation distance of 100m, which equates to 100 UAVs trajectories per km^2, the available capacity is presented in Figure 5-7, Figure 5-8 and Figure 5-9. In these 3D plots, the Cx, Cy represent the locations of congestion cluster centroids in the airspace, as shown along the x-axis and the y-axis, whereas the z-axis represents the traffic-flow capacity. The blue bar graphs represent the predicted traffic flow for each cluster region, per level. The magenta-colored plane, meanwhile, represents safe traffic flow for 100 UAV trajectories per km^2 (traffic-flow threshold). The green plane is actually the zero-values reference plane, plotted for better visualization. The peach bar lines indicate the available capacity, which is the difference between the predicted traffic flow and the traffic-flow threshold. A positive peach-bar height represents the available capacity of those UAV trajectories that can be accommodated in a 1 km^2 area. Negative peach bars, conversely, reflect non-availability of airspace in the relevant clusters.

It is evident from the three figures that, in the lowest-congestion zones, positive capacity bars outnumber their negative equivalents. The lower-level zone evinces

a mixture of both positive and negative traffic-flow capacity, while the medium-level congestion zones present more negative traffic-flow capacity. The higher and highest levels have either little or no positive capacity, and this reflects limited or zero availability of airspace for UAV operations.

In order to review the traffic flow distribution in the temporal domain, the ratio between the number of positive to negative capacity bars is presented as a capacity ratio (Table 5-5). This will highlight the efficiency of capacity in the three different scenarios that prevail from 9:00 am to 12:00 pm.

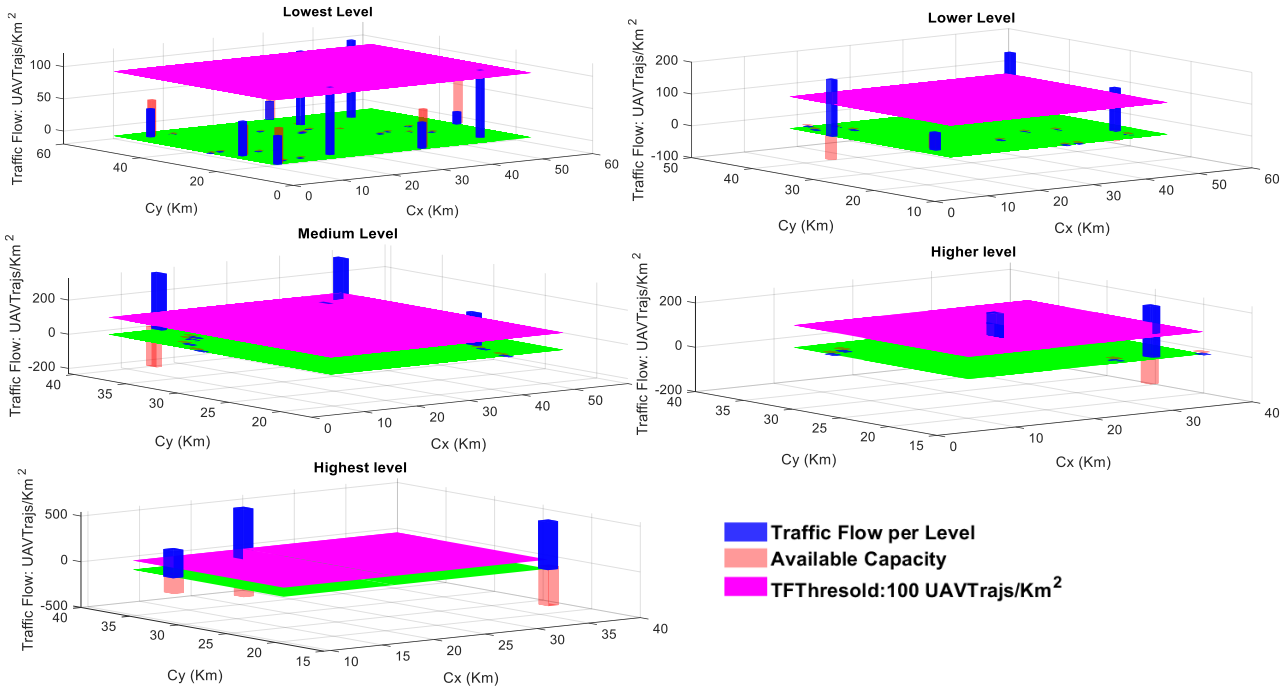

Figure 5-7 Traffic flow distribution and airspace capacity: Scenario 1

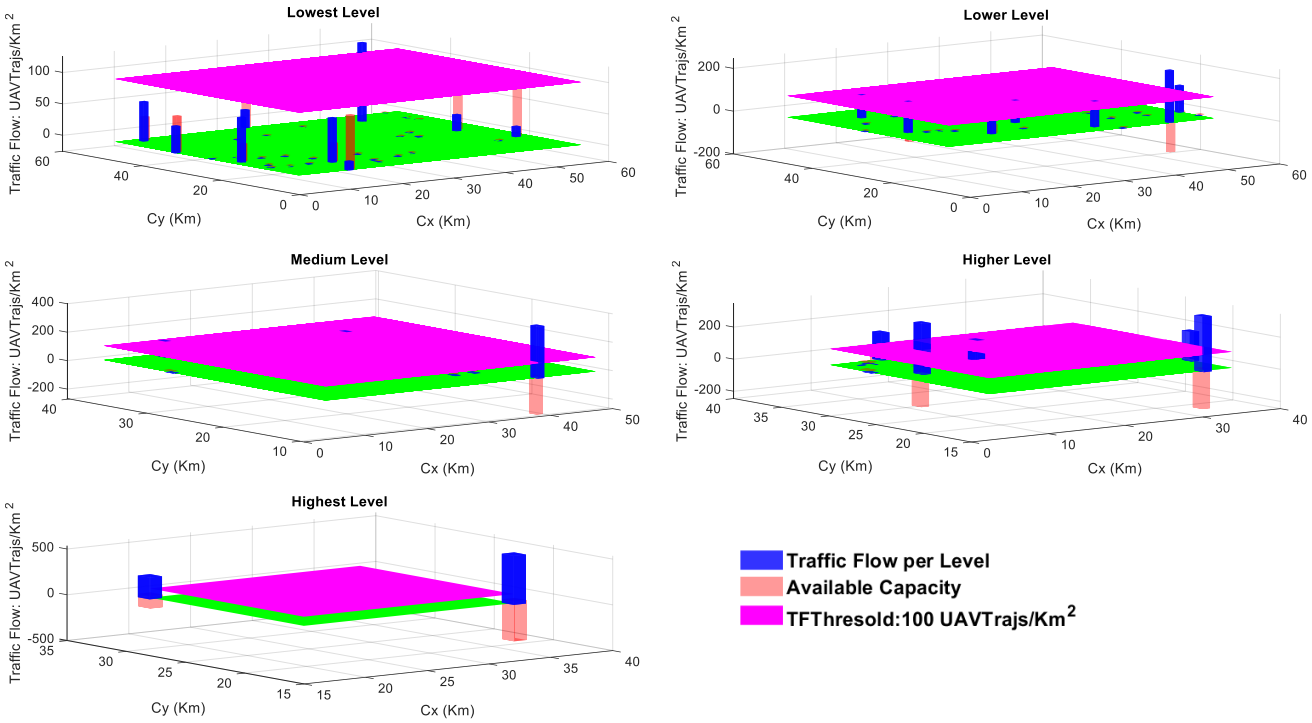

Figure 5-8 Traffic-flow distribution and airspace capacity: Scenario 2

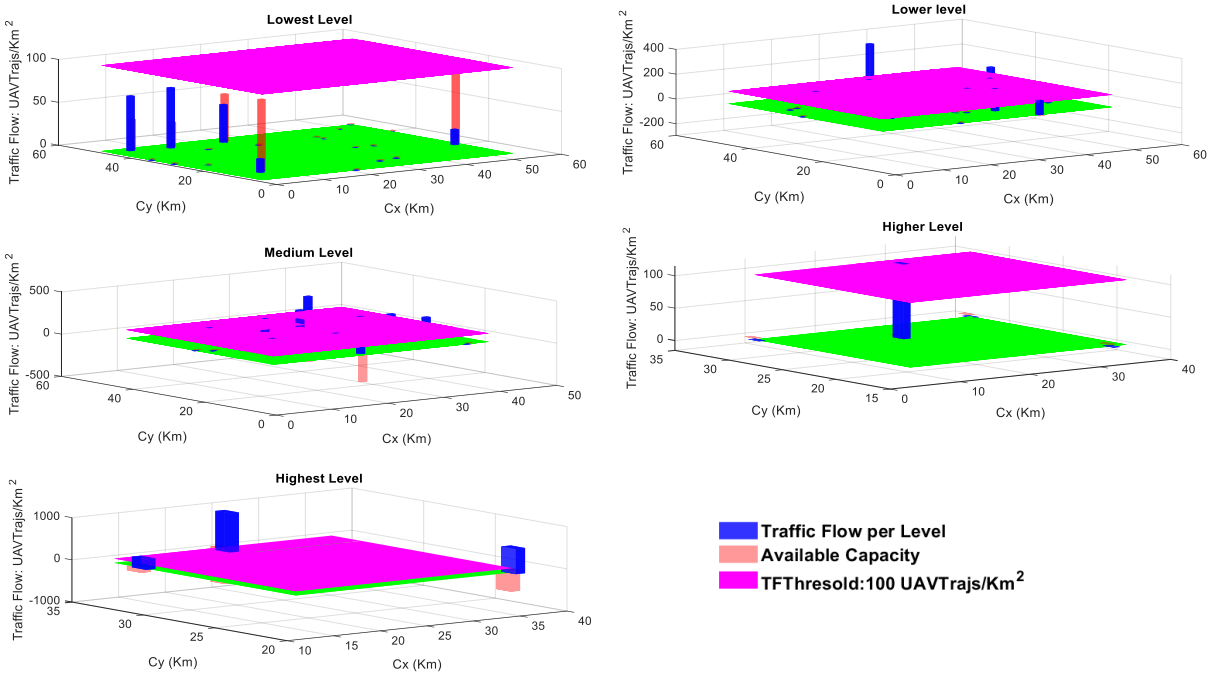

Figure 5-9 Traffic-flow distribution and airspace capacity: Scenario 3

Table 5-5 Capacity gain: a comparison

Congestion level	Capacity Ratio		
	Scenario 1 9-10 am	Scenario 2 10-11 am	Scenario 3 11am -12 pm
Lowest	6/4 = 1.25	8/1= 8.0	5/0 = ∞
Lower	2/3 = 0.66	4/6= 0.66	2/8 = 0.25
Medium	1/4 = 0.25	1/3= 0.33	0/9= 0.0
Higher	0/2 = 0.0	1/5 = 0.25	0/1 = 0.0
Highest	0/3 = 0.0	0/2 = 0.0	0/3 = 0.0

Table 5-5 indicates that the capacity ratio was higher in the lowest-congestion regions during adverse and extreme weather conditions (Scenario 2), rather than with static obstacles only (Scenario 1). This reveals that there were more opportunities to accommodate UAV trajectories during Scenarios 2 and 3 than during Scenario 1. The lower- and medium-congestion regions demonstrate that Scenarios 2 and 3 provide some capacity to accommodate UAV trajectories, although there is less scope for accommodation in Scenario 3. It was also noted that the regions with the higher and highest levels of congestion had very limited, or zero, opportunity to accommodate UAV trajectories across all three scenarios. It may thus be inferred that Scenario 3 resulted in more severe congestion, and this was consequently the most hazardous time zone in which to operate UAV missions.

The sections above demonstrate the traffic-flow distribution in different congestion subzones of the Bedfordshire airspace, followed by the provision of available capacity in these subzones. The preceding air traffic-flow analysis, in indicating the most appropriate regions for UAM operation (and non-available urban airspace) will clarify actual availability situations regarding UTM airspace. Moreover, this analysis enables the UTM operator to regulate and reconfigure UAV paths, based on graphs that, in turn, represent air-traffic hotspots. The analysis can also be used to mitigate congestion in predicted UAV-traffic hotspots, while suggesting appropriate congestion-free trajectories based on the UAV mission priority. The analytical model, indeed, seeks to reduce the workload

of the air-traffic controller by predicting congested areas in advance, and by facilitating appropriate action to prevent their formation.

5.4.2 Analysis and Design of Explainability for The DCM Advisory System

The present section introduces the UTM advisory-system framework. A transparency hybrid AI algorithm provides the foundation for the latter, and this algorithm facilitates a DCM framework, for process and solution, that is both flexible and resilient. The stringent operational demands associated with low-altitude airspace, especially in urban environs, are thus satisfied. A training data-generation element is included within the hybrid AI algorithm, whereby the decision-making model is both trained and optimized. Hence, systemic decision-making performance is improved. The capacity of the system to produce multiple, practicable solutions to problems of airspace congestion is indicated via a preliminary case study, described below. In order to allow end-users to understand the causes of particular behaviors, and to render the trained model itself more transparent, various readily interpretable visual and textual explanations are generated. Some explanations-by-example are provided, moreover, which illustrate the logic behind the decisions the UTM operator makes when indicating both the most appropriate regions for UAM operation, and available urban airspace.

Depending on criteria of explainability and accuracy, ML models may be placed in one of three principal categories, namely, Black Box, White Box, or Gray Box. White-Box models are those in which internal workings, logic and programming remain transparent. Consequently, the decisions they generate are readily interpretable. The most obvious example of the White-Box paradigm is a simple decision tree, but further instances are provided by Bayesian networks, linear-regression models, and Fuzzy Cognitive Maps [368]. The simplest models to explain are, generally, those that are monotonic and linear. Since certain fields, such as finance and medicine, evince a particular need for transparency, they are often associated with White-Box solutions [368], [369]. Conversely, while Black-Box ML models are frequently more accurate, their internal processes are opaque

and difficult to interpret. Hence, software testers, or stakeholders, may understand little more than the anticipated inputs and associated outputs of the model. The most widespread instances of such models are neural networks, either shallow or deep [369].

A Gray Box, as the name implies, combines characteristics of both Black and White [370]. As a compromise between the latter two, a Gray-Box solution seeks to reflect the main advantages of both, ultimately comprising a more effective, global composite model. Broadly speaking, the term Gray Box can be applied to any ML-learning algorithmic ensemble that presents both White and Black characteristics, and some forms of linear regression or neural network fit this category. Fairly recently, Grau et al. [368] constructed a transparent, accurate and interpretable predictive model, via a self-labeled methodology, that combined elements of Black- and White-Box models.

The recommendation system methodology employed in this study adopts a hybrid approach, integrating less-explainable (Black-Box) deep learning (LSTM) models for congestion prediction. This is followed by DBSCAN unsupervised learning, which is an explainable classifier, and finally by a rule-based decision tree (White-Box) approach to make a final recommendation. Metrics have been utilized to assess the overall explainability of the hybrid model employed in this study, with a focus on evaluating the transparency of the individual components. The Black-Box models are not transparent, due to a lack of clarity regarding their inner workings. By contrast, White-Box models evince observable and understandable behaviors. Scores have been assigned to the components of the hybrid model for the purpose of evaluation, as illustrated in Table 5-6.

Table 5-6 Methodology transparency scoring

Methodology	Scores
Black Box	0
Gray Box	0.5
White Box	1

In order to provide a more comprehensive assessment of the total explainability percentage, it is imperative to consider the dependency of each component. By recognizing the interconnectedness among the components, a more comprehensive comprehension can be attained regarding their combined impact on the overall explainability of the system or model. The component level development or modular approach is a well-known software system design methodology where every component is considered as isolated block with some Inputs→Process→Output and dependency is shown by the flow of information or data connecting the software components [371][372].

In this study, the explainability metric is established on the transparency of modules used in the overall framework that is based on Process aspect of a component. It is important to note that five major components have been identified in the XAI DCM methodology. These components are dependent on their predecessors, and their execution follows a sequential order. Currently it is assumed that each component is critical and equally dependent on its predecessor. However, the dependency factor could be made more explicit and a weightage may be assigned to make these metrics more robust in future studies. The five designated components along with their dependency and transparency attributes are shown in the table below. It has thus been calculated that the proposed advisory system is around 70% explainable.

Table 5-7 Advisory-system transparency

Advisory DCM Components	**Methodology**	**Dependency**		**Transparency Type**	**Score**
		Data Input	Data Output		
Congestion prediction	Deep learning (LSTM)	(x, y, Vx, Vy)	[x, y, Complexity metrics]	Black Box	0
Congestion levels assignment	Simple rule based	[x, y, Complexity metrics]	[x, y, %Congestion Levels]	White Box	1
Congestion subzones identification	Unsupervised clustering ML algorithm (DBSCAN)	[x, y, %Congestion Levels]	[x, y, Trajectory Clusters Per Congestion Levels]	Grey Box	0.5

Airspace capacity estimation	Rule based	[x, y, Trajectory Clusters Per Congestion Levels]	[Congestion Polygons, Cx, Cy, Area of Polygons , Traffic Flow, Capacity]	White Box	1
DCM decision	Decision tree	[Congestion Polygons, Cx, Cy, Area of Polygons , Traffic Flow, Capacity, New Missions Trajectory (x, y), Mission Priority]	[Allow/Disallow Airspace usage, Updated Capacity]	White Box	1
Total explainability percentage (%)		**70%**			

The proposed advisory system is a rule-based decision tree. It takes into account the information outlined above regarding each congestion level, and it checks whether capacity is available in the demanded cluster. Moreover, it also checks new, incoming UAV mission priorities. If capacity is available, then decisions regarding accommodation are made according to priority. If there is no capacity, the advisory system will suggest alternative available airspace for the UAV mission.

5.4.2.1 Rule-based Explanation for DCM Decisions

Any model that generates rules to characterize the data from which it is expected to learn is an example of “rule-based learning.” As a means of constructing knowledge, rules may comprise complex combinations of simple rules, or they may take the form of straightforward if/then conditional propositions. Fuzzy rule-based systems are also related to this model family, being designed for broader fields of action, and facilitating the construction of verbally defined rules for relatively imprecise domains. Since they can generate rules to clarify the basis of predictions, and since they are highly transparent, rule-based learners have often been deployed to explain more complex models [373]. They have also been widely used to represent knowledge within expert systems [342]. The rule-based methodology, as employed in the present study, unambiguously presents the

decision boundary that obtains between advice provided and contrasting advice: this takes the form of if/else statements. A pair comprises the local explanation, and this in turn contains (i) a logical rule, reflecting the decision-tree path that explains why a given decision output is classified as positive, and (ii) a counterfactual rule set, explaining why conditions should be modified, and an alternative decision should be registered. For instance, an explanation for the recommendation system used in a low-congestion zone could be provided : the rule (0%≤Congestion ≤20%, Available_Capacity > 0) Allow current airspace and the counterfactuals {Available_Capacity < 0}⟶Recommend free Airspace. The rule-based explanations for different cases are, meanwhile, presented in Figure 5-10.

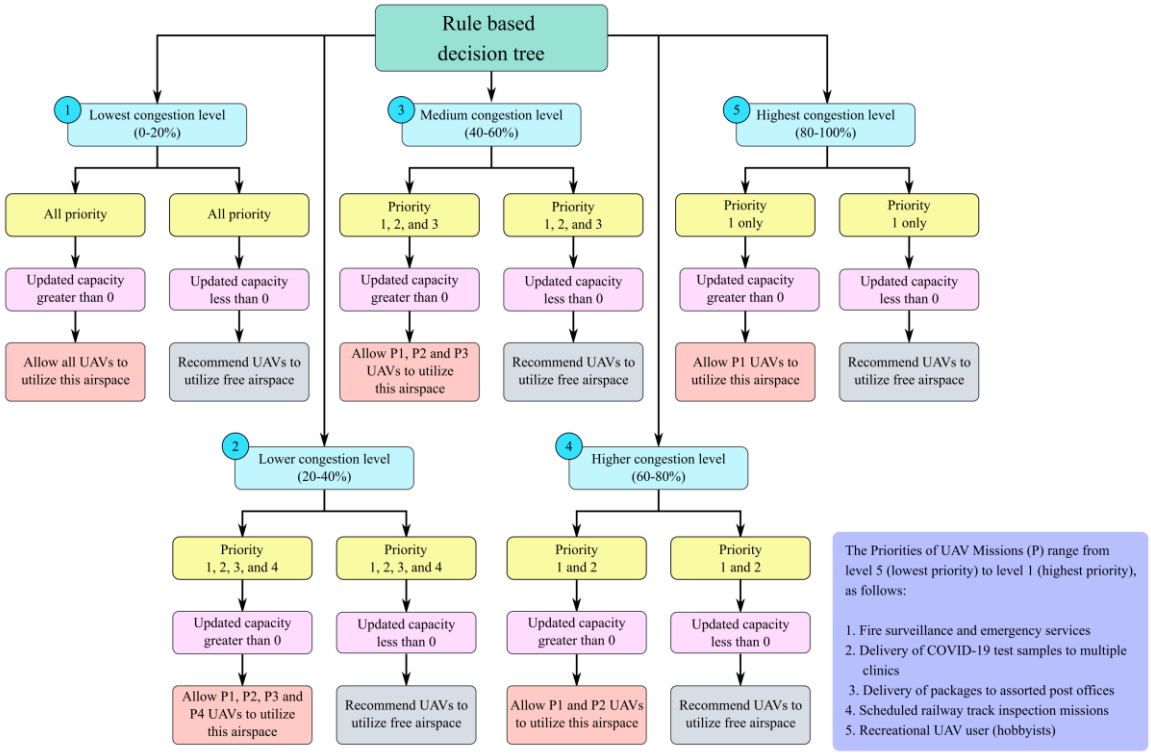

Figure 5-10 Rule-based decision tree for DCM services

5.4.2.2 Post-hoc Local Explanation: Techniques for Visual Explanation

The goal of visual-explanation techniques, in the context of post-hoc explainability, is to present model behavior in a visually comprehensible manner. Understanding may be further enhanced when these techniques are

supplemented by other methods. Still, visualization is generally regarded as the best means for introducing complex interactions between model variables, especially for users unfamiliar with ML modeling. Visualizations are popular because of their natural, intuitive nature [374]. Much scholarly analysis has been devoted to determining which forms of visualization are best suited to particular types of practice or application [351], [375].

To validate the results of the proposed decision-tree methodology, a constant number of incoming UAV trajectories (15 trajectories) has been assumed for the simulation in question. The priority level of the incoming UAV trajectory is treated as random, between priority levels 1-5. The results of the decision tree for all three scenarios are explained visually, and they are presented below in Figure 5-11, Figure 5-12 and Figure 5-13. In these 3D plots, the Cx and Cy are the locations of congestion-cluster centroids in the airspace, as indicated along the x-axis and y-axis, whereas the z-axis is the decision variable. The blue bar graphs, meanwhile, represent the updated capacity for each cluster region, per level. The magenta-colored bars represent the number of incoming UAV trajectories, while the yellow bars indicate the priority levels of incoming UAVs. The green plane, finally, is the zero-values reference plane, plotted for better visualization.

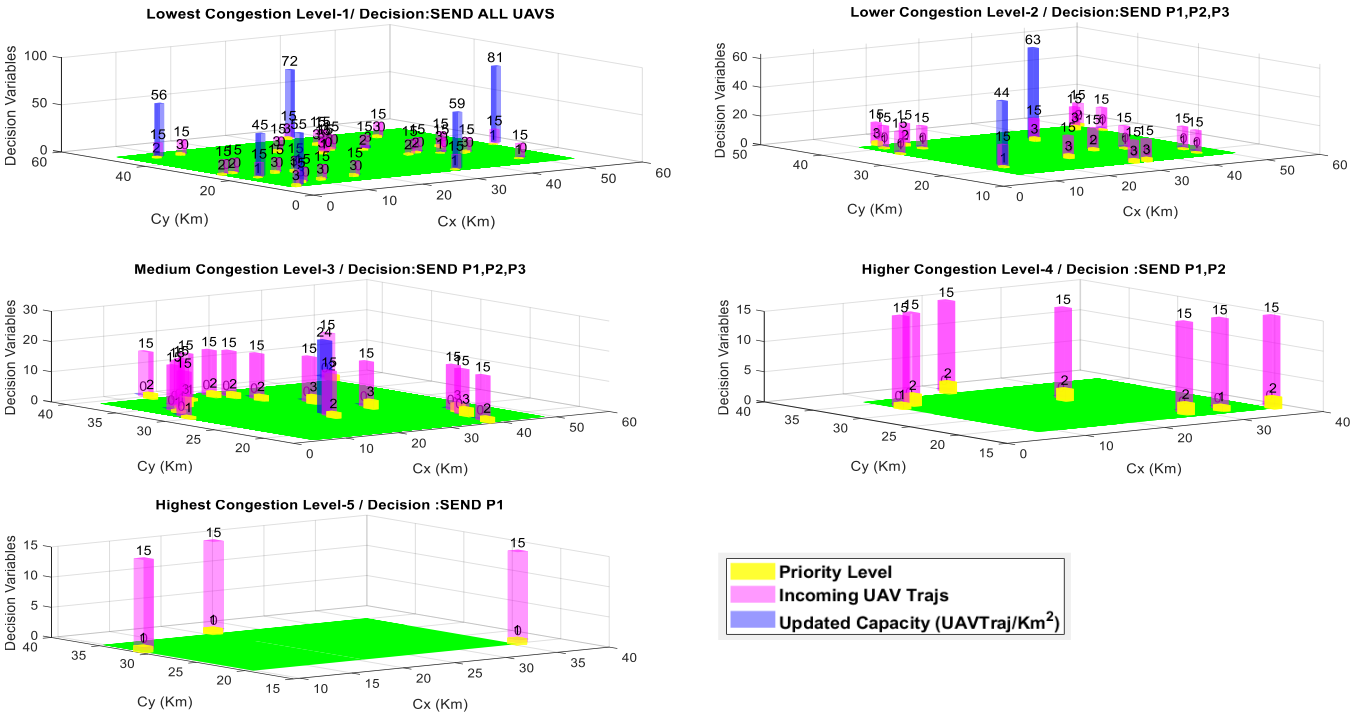

Figure 5-11 Decision-support results: Scenario 1

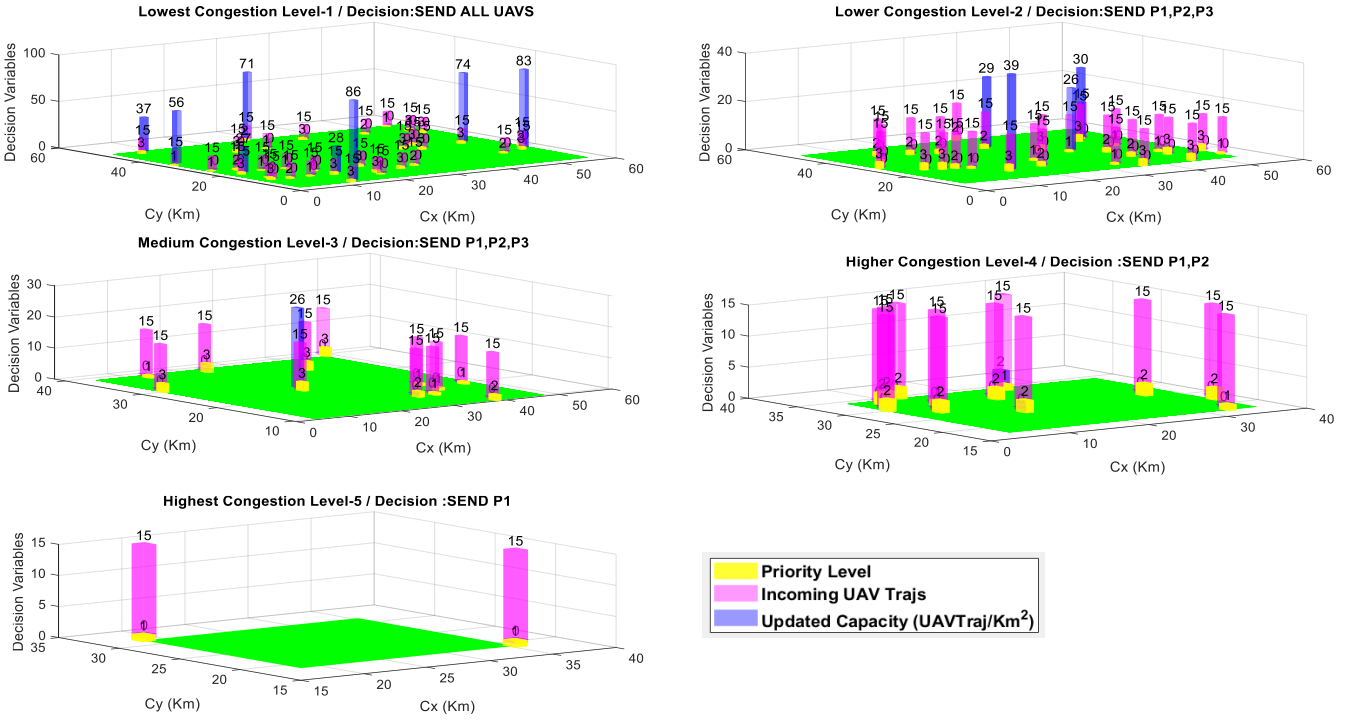

Figure 5-12 Decision-support results: Scenario 2

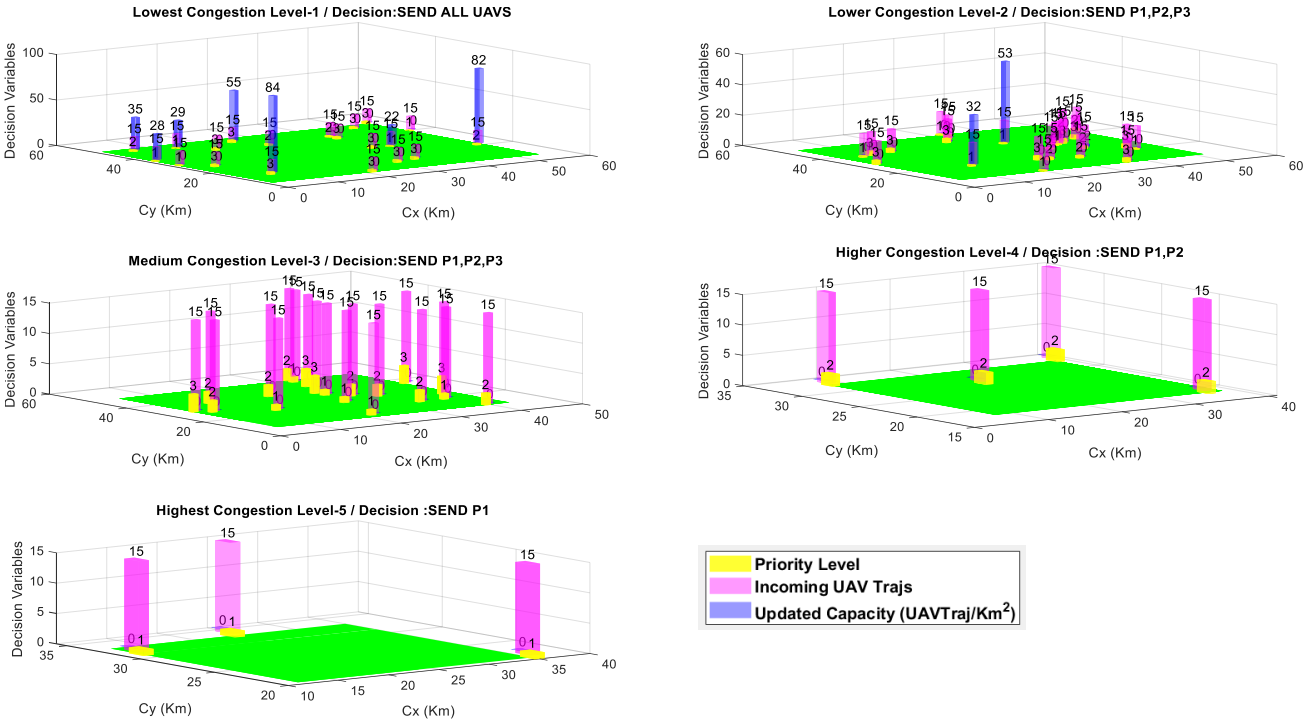

Figure 5-13 Decision-support results: Scenario 3

In the figures above, it may be observed that, if the updated capacity (blue bar) is positive, the UTM authority is advised to accommodate incoming UAV trajectories in that congestion region. This is in line with the priority-decision rule. If the updated capacity is negative, that particular cluster will be indicated as having no capacity; specifically, a zero value will be shown against that blue bar. Moreover, Figure 5-11, Figure 5-12 and Figure 5-13 show the advisory-system advice, in which the largest number of UAV missions with any priority should be planned for the lowest-congestion zones.

This advisory for the lowest-level congestion zone (Scenario 3) is illustrated with the help of one example (Figure 5-13). An emergency mission related to Covid-19 test-sample delivery (Priority 2) is planned in the vicinity of the lowest congestion-zone centroid (Cx=6.86, Cy=48.64) under extreme weather conditions. It is assumed that the planned mission has 15 UAV trajectory points, passing through this lowest congested zone. The updated capacity, as estimated for this sub-region by our recommendation system, is 35, and this is shown visually (blue bar) in Figure 5-13. The advisory system allows the fulfillment of this demand, bearing in mind the available capacity for this zone. A further

example is provided here, whereby an emergency fire-surveillance mission (Priority 2) is planned in the area close to the highest congested-zone centroid (Cx=24.33, Cy=33.68): this references Scenario 3, under extreme weather conditions. It is also assumed that the mission has 15 UAV trajectory points planned for this region. The updated capacity, as estimated for this sub-region by our recommendation system, is a negative number marked as zero (capacity overloaded), with no blue magnitude bar. The advisory system will advise the UTM operator to divert low-priority flights in this area to other areas, while creating airspace availability for current emergency operations.

The proposed advisory system thus helps support better airspace design, by providing a clearer picture of airspace congestion. This is supported by a characterization of airspace via five, lower to higher congestion levels. This will help UTM authorities achieve better flight management, based on historical or simulation-based data. Moreover, the advisory system suggests that missions planned through clusters with zero capacity should relocate, in order to use the freely available airspace and, thus, improve safety.

5.4.2.3 Post-hoc Local Explanation: Explanation by Example

In the case of explanations by example, the focus is on the extraction of data examples, with the latter pertaining to the results generated by a particular model. In turn, this supports a more lucid understanding of the model per se. Much like human behavior (when people strive to explain a certain process), explanations by example focus on extracting typical, or representative, instances. These, in turn, help users grasp the correlations of the model under analysis, along with its internal relationships [342].

Two illustrative examples can be presented to demonstrate the simulative and explanatory characteristics of the proposed model for a decision-support system. One example references the lowest-congestion zone, and - for completeness – the other concerns higher-congestion zones. In tandem with Figure 5-11 (Scenario 1) and considering the lowest-level congested zones (congestion < 20%), the inputs and output decision made by the advisory system can be elucidated to facilitate the inclusion of incoming unmanned aerial vehicles (UAVs)

en route to the specified zone or subregion within Cluster 5. For simplicity,15 new UAVs were assumed in Scenario 1. The incoming UAV is of priority level 2. The updated capacity is 56: since that is a positive number, more UAVs can still accommodate in this region before reaching the safe threshold of 100 UAV trajectories/km^2. According to the rules set for the lowest-congested region, UAVs with any priority can use this airspace. Consequently, the advisory system allows the UAV to utilize the airspace encompassed in Cluster 5. Capacity is thus enhanced without any compromise in terms of safety. This example is illustrated in Figure 5-14, where x signifies the inputs to the model M_{RS}, and y is the output decision suggested by advisory model. The exact input values (x1,x2,x3,x4) that resulted in output (y) are also presented in the callout diagram (Figure 5-14).

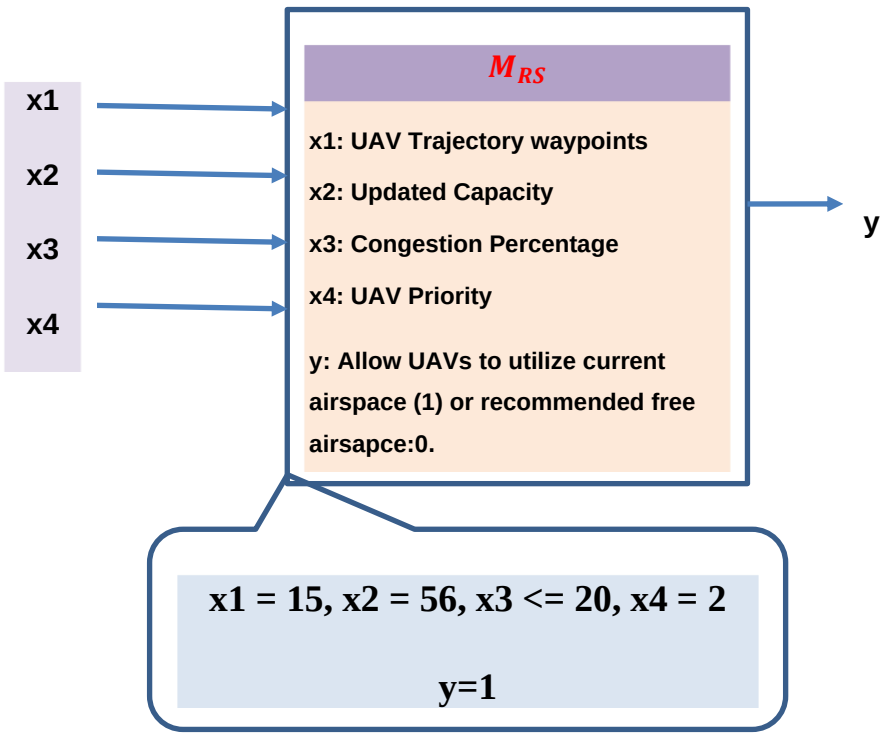

Figure 5-14 Explanation by example for Scenario 1, lowest congestion level

The second example simulates the decision taken in Scenario 1 for the highest-congested zone (80<congestion< 100%), implicating Clusters 1, 2 and 3 (Figure 5-15). 15 UAV trajectory points were planned for these regions, with all these UAVs engaged on priority level-1 missions. The updated capacity is a negative number, which indicates that the capacity threshold of 100 UAV trajectories / km^2

is being violated. As per the rules defined by the advisory model, UAVs are advised to avoid these highly congested regions by using alternate free airspace. The safety of the airspace is thus improved.

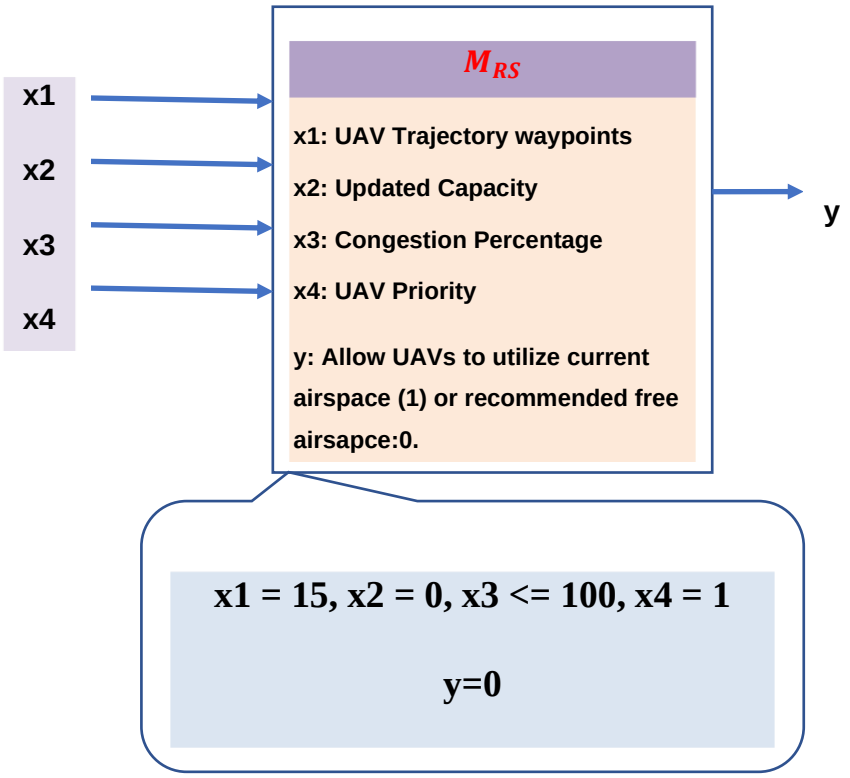

Figure 5-15 Explanation by example for Scenario 1, highest congestion level

5.4.3 Advisory-System Efficiencies for Capacity and Safety

This section introduces two metrics, namely, capacity gain and safety gain, that underpin the proposed decision-support system, while reinforcing its efficiencies for all three scenarios. The proposed advisory system "efficiency" is twofold in nature. First, it effectively utilizes the available capacity in congested cluster zones of UTM airspace, thereby maximizing capacity. Second, the advisory system pre-empts safety hazards by using updated traffic flow and capacity models, thereby excluding planned missions from some medium-, higher- and highest-level congestion clusters. Table 5-8 presents the advisory system efficiency in terms of capacity and safety gains. If a congested cluster is utilized, this means that one capacity gain is achieved. If entry to a congested cluster is prohibited, conversely, this means no capacity is gained, but rather, one safety

gain is acquired. The following equations are used to define the efficiencies of the advisory system in terms of capacity gain (CG) and percentage safety gain (SG):

$$E_{CG} = \frac{CG}{CG + SG} \times 100 \quad \textbf{(5-1)}$$

$$E_{SG} = \frac{SG}{CG + SG} \times 100 \quad \textbf{(5-2)}$$

Table 5-8 Advisory-system efficiency

Congestion level	Advisory System Efficiencies (%)											
	Scenario 1				Scenario 2				Scenario 3			
	CG	SG	ECG	ESG	CG	SG	ECG	ESG	CG	SG	ECG	ESG
Lowest	7	21	25	75	8	26	24	76	7	13	35	65
Lower	2	15	12	88	4	24	14	86	3	21	13	87
Medium	1	15	6	94	1	11	8	92	0	19	0	100
Higher	0	7	0	100	1	11	8	92	0	4	0	100
Highest	0	3	0	100	0	2	0	100	0	3	0	100

The advisory system efficiencies for capacity and safety are explained in Figure 5-16. The proposed advisory system presents more capacity-efficiency gains in the lowest- and lower-congested zones, and maximum safety gains in the higher- and highest-congested regions, for all three scenarios. The static-obstacles scenario (Scenario 1) and the adverse wind and rain scenario (Scenario 2) present similar efficiency trends in low- and medium-congested levels, but they evince dissimilarity for higher-order congestion levels. It may further be deduced that the advisory system exhibits maximum safety efficiency, and minimum capacity efficiency, for medium-, higher- and highest-congestion zones, when extreme weather conditions are experienced (Scenario 3).

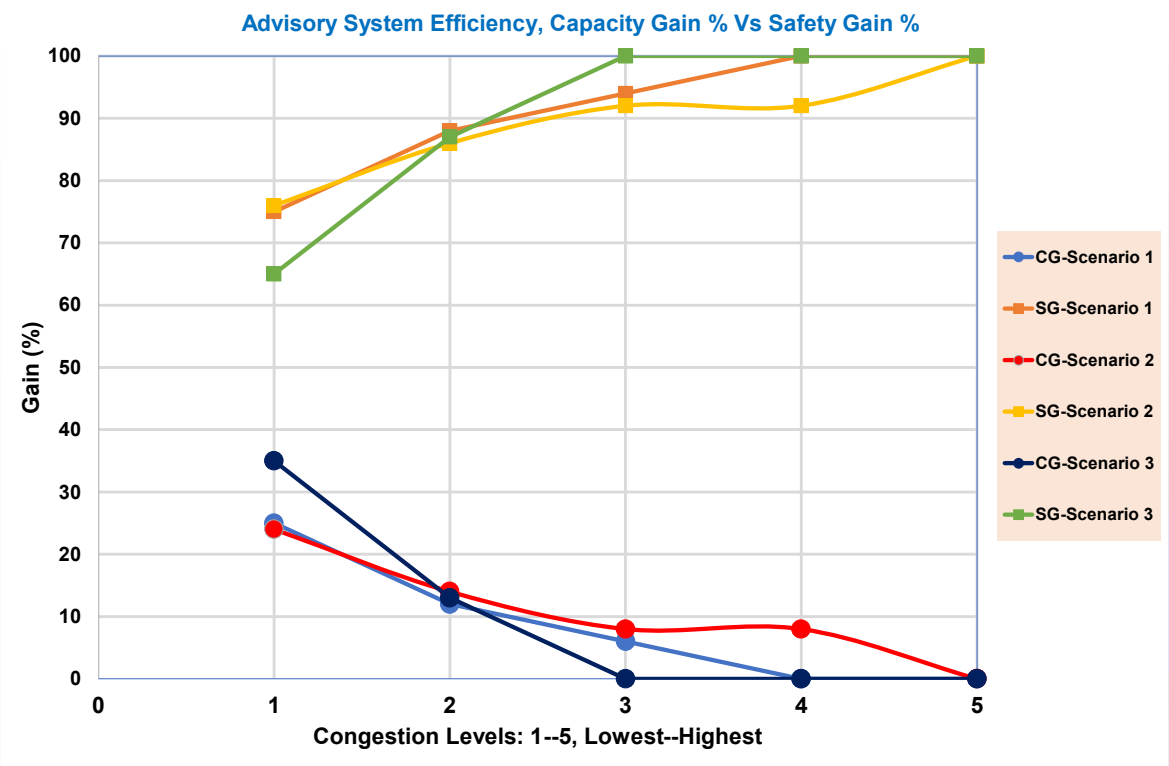

Figure 5-16 Capacity gain (CG) and safety gain (SG) for all three scenarios

5.5 Conclusion and Future Work

Compared with the traditional ATM environment, the difficulties associated with traffic-flow management in low-altitude airspace are markedly greater. For example, UAS operations involve traffic that is both heterogeneous and high in density; conventional, strategic planning methods for Air Traffic Flow Management cannot be adapted to such conditions. To complicate matters further, complex environmental variables are implicated in low-altitude urban airspace. These include physical obstacles, both static and dynamic, and the possibility of adverse weather. A wide range of AI/ML algorithms are characterized by non-transparent, Black-Box modeling paradigms, but this is unacceptable in the present case. Rather, decision support-system resolutions must be transparent, explainable, trustworthy, and readily understandable, and strategies must be developed to ensure this outcome.

The present thesis has proposed a decision-making system for UTM based on Explainable AI, and in particular, a Deep Learning Network (LSTM) combined with a rule-based decision tree, to determine optimal Demand Capacity-

Balancing solutions. This involves a hybrid approach using (i) Deep Learning for the precise prediction of congestion, (ii) DBSCAN data analytics for detecting physically congested sub-region cluster patterns, and (iii) rule-based decision logic for understandable UTM decision making. This hybrid approach results in an intelligent, comprehensive and understandable decision support-system model. The approach also combines the modules of flight planning, airspace configuration and demand capacity-balancing optimization.

The congested-clusters pattern analysis for all three scenarios (including static NFZ, as well as adverse wind, rain and severe weather conditions) reveals that congested-cluster areas are broadly spread across the entire Bedfordshire UTM airspace, as regards lower- and lowest-level choked areas. The medium-level congested areas appear slightly denser and less widely spread, compared to those of the lower- and lowest-level congested regions. The sub-regions with higher and highest levels of congestion show more intense grouping, and more centralized regional locations, as compared with other congestion-level groups. Upon further investigation, it was observed that the availability of UTM free airspace decreases from 42% to 23% within the first three hours of UAV operations. This decline can be attributed to adverse weather conditions such as wind, rain, and extreme weather fronts. The overall UTM congested airspace in these regions also increases during these hours of operation, from 34% to 49%, respectively.

The study of traffic-flow capacity reveals that, under a safe separation distance of 100m, the lower- and lowest-congestion zones in the Bedfordshire area present ample capacity for future airspace operations. Indeed, the medium-congested zones also offer some available capacity. The higher- and highest-congestion zones present no capacity, however, and thus no airspace availability for UAV operations. The capacity-ratio analysis further reveals that, for lower-congestion zones, adverse wind, rain and extreme weather conditions still allow additional capacity for airspace operations. Under scenarios of extreme weather conditions, by contrast, the higher- and highest-congested zones do not present

any airspace to accommodate UAV missions. Such times, in fact, are the least propitious for operations in these zones.

One may conclude, from the proposed visually explainable advisory system, that more UAV missions (of any priority) may be planned for the lowest-congestion zones for the three hours in question (9:00 am -12:00 pm), within Bedfordshire airspace. Moreover, the advisory system shows capacity-efficiency variation from 25% to 35% in the lowest-congested zones, while this falls to almost zero for higher-order congested regions during the three hours of UAV operation. The safety efficiency varies between 75% and 65% in the lowest-congested areas, but this reaches almost 100% in their highest-congested counterparts. It may further be deduced that the proposed advisory system demonstrates greater safety efficiency, but less capacity efficiency, for medium-, higher- and highest-congestion zones under extreme weather conditions. The proposed model permits the UTM operator to regulate and reconfigure UAV paths, based on complexity predictions representing air-traffic hotspots. The model can also be used to mitigate congestion in predicted UAV traffic hotspots, while suggesting appropriate conflict-free trajectories. The advisory-recommendations system would help UTM systems to off-burden the manual decision-making process via visual data analytics, and via suggestions made by our proposed model for safe and optimal-capacity airspace environments. The proposed advisory system is also verifiable through explanation by example, and this should increase the confidence of UTM authorities in adopting the proposed hybrid methodology. Indeed, the scoring-based metrics evaluate the explainability percentage for the proposed system at about 70%.

In future endeavors, there is an intention to broaden the application of the proposed interpretative framework to additional domains or areas. An alternative approach could involve leveraging other AI techniques , by deploying other varieties of AI, such as SHAP, LIME, global and/or local explanatory methodologies. Additionally, the application of the aforementioned work is being considered for other regions of UK airspace that exhibit a more varied array of airspace configurations and are expected to experience a significantly higher volume of UTM traffic in the future.

6 GENERAL DISCUSSION

6.1 Introduction

The present chapter aims to explore the pertinent findings of this thesis, highlighting areas of strength and how the findings contribute to the expansion of the literature base. It will also explore prospective limitations and how these could be improved upon in future research.

6.2 Simulation Development

Simulation-based tools play a crucial role in research programs related to Unmanned Traffic Management (UTM) studies. This thesis highlights the significance of requirements elicitation and analysis in the development of such tools, specifically focusing on the requirements for UAV transport models. It is widely acknowledged that gathering requirements and creating use case models are essential to ensure that a system fulfills the needs and expectations of its users and stakeholders in software development for these tools. Therefore, a use case model (Section 2.2.1) is derived based on a set of requirements for an idealized transport system intended for drone-based delivery service missions. These requirements are compiled in accordance with CAA regulations [45] , UAS Traffic Management (UTM) Concept of Operations [46] ,SESAR U-Space [47] and the SORA methodology [74] . The key requirements in the development of a UAV transport model are as follows:

1) Optimal trajectories for missions that are safer, time- and energy-efficient are planned.
2) Energy consumption is considered based on UAV models and types.
3) Different types of UAV missions (linear or area) are taken into account.
4) Payload weight factors are considered.
5) The priority of different UAV missions is also taken into account.
6) Uncertainties in the airspace environment, such as weather constraints, NFZ and dynamic obstacles, are to be aware of.
7) Separation management, collision avoidance, and deconfliction strategies are implemented for the missions.

8) Ensuring adherence to relevant rules and regulations is crucial when using drones for delivery purposes.

The use case model provided in this regard serves as a foundation for designing and implementing a UAV transport system simulator and practical tools to aid UTM authorities. This idealized transport model can be readily customized and adapted to create case study models with varying assumptions, as demonstrated in different case studies conducted during the course of this research. Furthermore, the design template offers a framework that can be embraced by a broad community, including academics and practitioners alike. The use case design model presented here is adaptable and scalable, allowing for the incorporation of emerging concepts and UTM regulations through extensions in UML design semantics.

The development of UTM simulators, in turn, reflects two distinct approaches, as noted in (Section 2.3) : one relies on open platforms and sources, while the other favors proprietary solutions. Certain European and US initiatives, the proponents of which have long experience with ATM, incline towards their own technological assets, and this is also true of some private enterprises, such as Airbus and OneSky. The second approach, i.e., the use of open sources and open platforms, is favored by academia and smaller research units. Thus, without heavy expenditure on tools or simulator-platform construction, these actors can focus immediately on UTM-oriented research. Flying large quantities of drones in one section of airspace, meanwhile, is costly in terms of money and time; thus, UTM-systemic research tends to be undertaken via simulators. To assess the performance of traffic-management models for small drones at low altitudes, however, NASA maintains that dedicated simulations are required.

The present study sought to improve on current UTM simulators, with particular emphasis on the key performance indicators (KPIs) identified in the literature review (flexible, scalable, support real-time 3-D visualization, addresses challenges in UTM systems and easy to use/modular/free/portable across platforms). A key strength of the present study is that the simulation parameters

were developed via MATLAB – this means that the model is both free and open source. It is supportable via several operating systems (e.g., Windows, MacOS, and Linux), and it is scalable, interoperable, extendable, and portable. Given that the purpose of the simulation is to improve safety of airspace, these criteria ensure that it can be widely utilized and thus have the most comprehensive impact on the field. It is also able to seamlessly navigate the true complexity of airspace, making it more ecologically valid - this simulation can schedule any number of flights within each scenario and can accommodate random start and end times for missions. Further supporting the need for real-world application is its ability to incorporate the effects of static and dynamic obstacles, weather conditions, and random UAV trajectories of flight hobbyists (i.e., recreational users). It is essential that simulators can account for such variables, given the projected increase in demand for UAVs.

In this simulation, both dynamic and static airspace structures (including dynamic airfields, static NFZs, and recreational areas) are supported by the dynamic airspace configuration module. Extreme weather fronts are, meanwhile, and Wind turbulence is accommodated by the adverse-wind model, which also addresses levels of precipitation. The model supports dynamic and static geofencing, alongside other concepts of UTM airspace-safety management as indicated in various state and private initiatives (e.g., SESAR U-SPACE, Airbus, Amazon, Uber, and NASA-UTM). The developed simulation framework currently supports several deconfliction strategies, notably rerouting, ground delays, and hovering. The present approach is scalable, and it may be extended to comprise more sophisticated real-time strategies in the future.

To improve the present model further, it would be beneficial to develop an object-oriented simulation-framework design, as this would be better able to accommodate modularity, complexity, and maintainability. Unfortunately, the project was limited to the 2D (x, y) cruise phase of UAVs, meaning the ascent and descent phases require further analysis and visualization. This would be a valuable area for further research. From a safety-focused perspective, to accommodate sensor-based deconfliction algorithms such as sense-and-avoid, the inclusion of sensor models, navigation, and cooperative communication are

needed. Additional issues that need to be modeled in the simulation framework include boundary management, communication failure, and dynamic airspace configuration. In the future, better strategies and solutions for conflict identification and intelligent resolution may be provided by machine learning that could be incorporated into the current simulation. Furthermore, to mitigate the risk of inter-vehicle collisions in civil airspace, the Collision Detection and Avoidance System (CDA) may be employed by this simulation framework. Finally, results may be rendered far more realistic by the development and integration of a complex, dynamic modeling paradigm for UTM meteorological support.

The simplicity of the present model is a key benefit. In this work, particle swarm optimization (PSO) was utilized to provide optimal paths from a UAV service start point to the delivery point. PSO has fewer adjustable parameters, is easier to implement, and is more effective at converging towards a holistic optimal solution. In the future, however, it will be necessary to include more sophisticated approaches to path-planning improvement, i.e., approaches rooted in machine-learning technologies.

The literature base strongly indicates a likelihood for UAVs to become involved in different services, including surveillance, deliveries, farming and photography, and therefore anticipates huge expansion. The present study emphasized such variability in UAV application, considering diverse missions, and a variety of drone applications, from package transportation and railway inspection to fire surveillance and medical delivery. It also drew on the prioritization recommendations of NASA UTM and U-Space, as in true reality, different UAVs will require different prioritization levels. The mission-planning module is a key asset of the simulation; however, it is important to acknowledge the very low altitude of UAV flight paths, meaning that the 3D visualization of geoinformation is essential for UTM simulators. In this regard, future work may explore the potential of various 3D-visualization platforms, thus facilitating the insertion of 3D objects into virtual environments. Promising candidates may include Bing Maps, Google Earth, and AirMap.

6.3 UTM Airspace Traffic Pattern Characterization

The present study contributes significantly to the field through its successful characterization of UTM traffic-flow patterns. This is particularly helpful in airspace monitoring, performance assessment, airspace design and traffic-flow management. The novelty of this work lies in the incorporation of parametric uncertainties that may hinder UAV mobility, such as dynamic airspace structures, weather conditions and highly prioritized emergency operations. The utilization of DBSCAN as the algorithm for trajectory-data clustering is of great value to this model, and future model developments, as it enables identification of patterns of independent missions among the swarm of random, noisy recreational flights. DBSCAN is efficient in highlighting outliers and irregularly shaped clusters. Consequently, even when atypical trajectory profiles are apparent, core trajectory patterns may be identified. Moreover, DBSCAN can obviate the necessity of specifying cluster numbers in advance through detection of non-convex clusters.

Traffic-pattern identification has scope for further improvement beyond the limits of this research. The findings of this study indicate that DBSCAN-based unsupervised learning methodologies can identify and detect approximately 82% of UAV missions under favorable weather conditions and without any flying restrictions. However, it can be observed that mission detection capability decreases by up to 23% when there are variations in weather conditions and dynamic airspace structures. To address this issue, exploring other techniques such as HBDSCAN, SVM, and deep learning methods is suggested to improve the identification capability of missions in the presence of these constraints.

Similarly, there is potential for further improvement in characterizing traffic patterns beyond the scope of current research. In this work, characterization metrics such as DFC, DTCM, and TTCM were proposed, which helped to observe that some of the UAV missions in the Bedfordshire area exhibited unpredictability, resulting in a trajectory deviation with a maximum spread of approximately 8.6 km. It can also observe a shift in the mission centroids, with a maximum deviation of 1.88 km in horizontal directions and 1.55 km in a southerly direction. In addition, there is a significant increase in mission ranges, which span from 20 km

to 79 km, resulting in a worst-case increase in mission completion times of 14 to 49 minutes. Since the analysis of mission trajectory dispersion is limited to lateral or horizontal deviations during the UAV cruising phase only, In future studies, it would be beneficial to analyze the vertical mission deviations that occur during descending and ascending maneuvers of UAVs and develop new metrics to address this aspect.

These research findings should also, in due course, be applied to other UK regions, particularly those with higher volumes of UAV traffic or those that exhibit more heterogeneous airspace configurations and different weather conditions. There is also scope to enhance congestion estimation further by taking into account UAV system dynamics while implementing an AI framework capable of predicting airspace congestion patterns. This will facilitate the use of historical UAV traffic patterns to classify airspace regions as either dangerous or safe.

6.4 Traffic Flow Prediction for UTM Application

The present study was also successful in predicting UTM airspace traffic flow using a Deep Learning model that combines LSTM and 1-D convolutional neural networks (1D-CNNs). This can be beneficial for airspace monitoring, congestion forecasting, and traffic flow management. The key contribution of this work is the integration of cutting-edge machine learning techniques that incorporate UTFM with an intrinsic complexity metric, considering the Linear Time Variant (LTI) dynamics of UAV systems for predicting air traffic congestion. The proposed model is designed to adapt to the dynamicity of the UAV's operational environment (UAV's states) while considering factors such as shared airspace and weather-related interferences. The results show that the proposed model achieves the smallest RMSE value in all scenarios compared to other approaches. Specifically, the prediction error of the proposed model is 8.34% lower than the shallow neural network (on average) and 19.87% lower than the regression model on average. This confirms the hypothesis that combining the advantages and features of various algorithms and models (i.e., 1D convolutional, encoder-decoder LSTM and Linear dynamical system) affords superior problem resolution.

In existing works, the CNN model has been employed to predict density in an urban area. Nonetheless, one of the major disadvantages of CNNs is the excessive cost of computation. The present study proposes a DL model comprising LSTM and 1D-CNNs. This will have the dual benefit of enhancing deep neural network accuracy and saving time.

There is scope to expand on the present model to predict longer horizons and multiple traffic states given UAV trajectory data that covers a long duration and a wide range of states. In addition to longitudinal and lateral positions of the UAV, it may be possible to add orientation angles (pitch, roll, yaw) and angular velocities (pitch rate, roll rate, yaw rate) to improve the accuracy of estimations. Moreover, in order to test the robustness of the proposed model, more data from different cities representing various airspace structures is required to validate the variation effect on the prediction accuracy. Regarding computation of the complexity metric, the model used 3D prediction of the traffic flow. Benefit could be drawn from research on 4D trajectory scenarios (calculating the longitude, latitude, altitude, and time) as 4D trajectories help to improve the level of traffic flow prediction, thereby helping to avoid potential flight conflicts and enhancing air traffic safety. Finally, although the proposed method of traffic congestion prediction can capture the complex spatiotemporal dependencies in the current air traffic network, it does not consider changes in the airspace structure and therefore requires retraining following any change. Graph neural networks (GNNs) represent a promising candidate for traffic forecasting problems due to their capacity to capture spatial dependency, which is represented by non-Euclidean graph structures.

6.5 Explainable AI for DCM Services

The primary innovation of this study is the creation of a hybrid model, which incorporates a Deep Learning model using LSTM, iterative DBSCAN and a rule-based decision-making approach to improve the interpretability of UTM demand and capacity management decisions.

From a real-world or practical implementation perspective, this element of the model has the greatest impact. The visualization of airspace congestion allows the UTM operator to monitor and adjust UAV paths using graphical

representations of air traffic hotspots. This helps in managing congestion in anticipated UAV traffic hotspots and suggesting alternative routes based on the mission priority of the UAV. Additionally, this analysis aims to reduce air traffic controllers' workload by predicting congested areas beforehand and taking appropriate measures to prevent their formation.

Congestion data is attained by running a LSTM encoder-decoder prediction model based on the principle of the linear dynamics of UAV systems, which simulates a better picture of the airspace congestion keeping account of all the spatiotemporal dependencies. The resolution, however, is compromised and may be improved by considering more than five congestion levels (e.g., 10 or 20). This would add computational overhead on behalf of DBSCAN clustering for identifying the airspace sub-regions marked with these congestion levels. The model is also limited in that the visualization of congestion patterns will become complicated for the UTM authority in such cases. It may be beneficial to consider generating congestion data using other techniques, such as KDE, in future work.

Demand and capacity management was designed using a rule-based decision tree that takes into account congestion levels, available capacity and the priority of new planned UAVs to operate in the airspace. This aids better capacity management and also improves safety. One area where this study is different from other models is that it uses a combination of rule-based decision trees and deep learning. Rule-based decision trees are less accurate compared to deep learning methods; however, ML models are less explainable. This tradeoff was addressed by providing a hybrid of both methods and introducing novel explainability metrics on the concept of black-box, white-box, and gray-box model interpretations. It is thus assessed by applying the above metrics that the proposed advisory system is about 70% explainable. However, this method is new, established on quantitative scoring, and may be improved using a more comprehensive grading mechanism that may assign transparency grading at the functional level within each advisory component.

The established model showcases the explainability and comprehensibility of the proposed advisory system through rule-based explainability and two post-hoc explainability techniques based on visual explanations and explanations by example. While visual explanations provide a natural way to demonstrate the understandability of any model, they may not be adequate for visualizing complex multidimensional data. Similarly, explainability by example may not cover the entire scope of the models, given the infinite number of possible examples. To address these limitations, future work could explore the use of more comprehensive local explanation models, such as LIME and SHAP, which are freely available and do not interfere with the algorithm inside the black box. Additionally, these tools support multiple explanation plots and can quickly implement and explain tree-based models. By adopting X-AI tools, it is possible to build trust and understanding between UTM authorities and AI-based decision support systems, thereby reducing the workload of UTM operators.

Whilst there are many strengths to the devised model, the development of specific modules remains in a stage of infancy. In terms of flight planning management, for instance, the question remains of how to incorporate high-fidelity trajectory computation (e.g., 4D trajectory with weather impact) and collaborative decision-making mechanisms (for fairness concerns) into the DCM service. It is necessary to leverage a risk-based flight planning tool that fully incorporates the weather effects in planning the trajectory. Regarding airspace management, it is necessary to appraise the fundamental structure, along with the capacity prediction, dynamically modification and configuration adjustment. Furthermore, demand-capacity balancing could appraise potential solutions to non-nominal uncertainties and how to associate the solutions with flight planning and airspace management. Some promising learning-based methods could also be explored.

The present model might provide a foundation for future studies, if researchers interrogate rerouting for the avoidance of congestion (this being common in commercial aviation, and therefore a likely option for UTM). To mitigate delays, UAV flights might have their routes altered prior to takeoff, or flights could be

rerouted to bypass areas of congestion. Commercial aviation already evinces approaches such as trajectory option sets (TOS), whereby a group of specific trajectories, alongside tolerable delays, is submitted for each flight: hitherto, there is meager research on this model as an option for UTM. The latter's optimization inputs, moreover, might be airspace volumes reserved by operators, rather than particular trajectories per se. In terms of how to manage airspace-request constraints, airspace reservations, equitable treatment for larger and smaller operators, and other requirements, several pertinent research questions remain. With a view to enhancing fairness between earlier and later fliers, scholars have previously suggested limitations on the advance period for resource reservation or allocation [65]. Still, the efficacy, design, and fairness of such paradigms remain challenges, in terms of both realization and assessment.

In future studies, various types of UAVs should be taken into account since they have distinct requirements for safe separation distance due to their diverse flight characteristics, such as speed, endurance, altitude capabilities, and weather resistance. This can affect the areas where UAVs are permitted to fly and how they interact with other airspace users, which could potentially impact the estimation of airspace capacity.

7 CONCLUSIONS AND FURTHER RESEARCH

7.1 Addressing the Research Aim and Objectives

The integration of unmanned aerial vehicles (UAVs) within the existing air traffic management (ATM) system has encountered challenges due to the high volume of UAV traffic. To address these challenges and optimize airspace capacity, artificial intelligence and machine learning technologies have emerged as promising solutions. This study aims to provide practical insights into the potential and opportunities for machine learning (ML) in integrating UAS operations into different airspace classes. The proposed framework and methodologies have been implemented and tested, offering tangible outcomes that can serve as a foundation for future development in the UTM domain. Additionally, the study recognizes the widespread deployment of UAVs in smart-city scenarios and the significant interest from multinational companies and various sectors. However, concerns regarding safety, security, and the management of urban airspace remain unresolved obstacles. The current ATM system is ill-equipped to handle the expected increase in UAV traffic, highlighting the need for innovative approaches and solutions.

Motivated by these challenges, the present study has identified methods to overcome these limitations and therefore improve airspace safety and capacity. This research aims to investigate the challenges and opportunities in the field of data analytics for effectively delivering essential UTM services. This will involve the development of a data-analytics framework capable of analyzing historical UTM data to predict air traffic flow patterns and generate recommendations to enhance airspace safety and capacity. The aim is achieved by fulfilling the following objectives:

The lack of historical UTM traffic data necessitates the generation of data for analyzing UAV traffic flow patterns. This can be achieved through a simulation framework that takes into account the requirements of key stakeholders. The framework is depicted as an idealized use case transport model for drone-based delivery mission. Most researchers have focused on predicting future traffic

density based upon historical data. They assumed a static environment with fixed vehicle start and destination points, as well as fixed airspace constraints regarding no-fly zones (NFZ). A static environment is not directly applicable to UTM systems due to the dynamicity of the UAV's operational environment. The present study adds value to the research field by moving away from the static environments assumed in previous research, such as fixed points and fixed airspace constraints, while ignoring weather conditions. Emphasis was placed on practical scenarios that consider uncertainties arising from adverse weather conditions, as well as static and dynamic obstacles found in urban environments. These scenarios take into account the requirements for an idealized transport model in accordance with CAA regulations, UAS Traffic Management (UTM) Concept of Operations, and SESAR U-Space. However, it is acknowledged that this model is still in its early stages of development. The study has demonstrated its viability, and it is now important to consider further advancements to the model to enhance its utility.

The study also developed a data analysis framework for UTM applications that could identify and characterize UAV trajectory patterns under variable airspace structures and environmental factors. The scope of analysis in all previously mentioned studies is constrained to air traffic management and does not take into account the challenges in the UTM context. Furthermore, in operational performance analysis, the influence of airspace structure is generally disregarded in historic research. Moreover, few studies have examined the effect of weather conditions in defining air traffic flows. To compare and characterize air traffic performance at various scales, the present study considered the entire flight trajectory, in contrast with previous works, and includes terminal maneuvering areas and enroute trajectories for UTM airspace. The data processing framework, which comprised density-based spatial clustering of applications with noise (DBSCAN), identified significant deviations in mission patterns with almost 82% confidence levels - this has the potential to significantly impact the coordination of UAVs, and therefore safety. Moreover, the UTM traffic flow characterization is conducted by three novel characterization parameters mainly Distance from

Centroid (DFC), Distance to Complete Mission (DTCM) and Time to Complete Mission (TTCM). Moreover, the use of Kernel Density Estimation in this study is of significant value and will seamlessly aid operators to direct recreational fliers or low-priority missions towards areas of lower congestion, meaning high-priority flights have greater airspace available and can move efficiently through this.

The penultimate objective of this study was to develop a deep learning-based model to predict air traffic congestion, with the purpose of addressing issues relating to UAV traffic management. The study proposes a learning-based model to predict air traffic congestion over a period of three minutes based upon existing machine learning techniques. The use of the model aims to address issues related to UAV traffic management and focuses on predicting the congested region, considering air traffic uncertainties instead of addressing the conventional issues of trajectory prediction or conflict detection and resolution. It has been demonstrated that the proposed LSTM network outperforms other more mature approaches, exhibiting a significantly smaller (RMSE) compared to the shallow neural network (8.34% lower) and the regression model (19.87% lower).

Accurate estimation of airspace capacity is critical for a safe and efficient air-transportation system. Traditional approaches, which highlight controller workloads, deploy various methods for assessing airspace complexity. These methods, however, merely address controllers' operational conditions. In the real world, such model-driven approaches fail to capture true airspace capacity, since they fail to address other important variables, such as weather. The construction of highly automated UTM systems, making use of artificial-intelligence (AI) algorithms, is technically feasible. Intelligent algorithms, however, are often characterized by inexplicability and opacity, and this restricts their practical application. This study proposed a tailored solution to the needs of demand-and-capacity-management services address the trade-off between explicability and performance. In other words, it will generate an intelligent system that will be both explicable and reasonably comprehensible. The results show that this advisory

system will increase UTM airspace availability by more than 23% while possessing an explainability attribute of 70%.

This research framework presents an opportunity to efficiently and effectively impact airspace safety and capacity, enhancing the industry's ability to cope with the growing number of UAVs and the need for integrated manned/UAS Traffic Management (UTM).

1. The data analysis carried out here supports risk analysis and improves trajectory planning in different airspace regions. Moreover, it considers all dynamic parameters (such as extreme weather conditions, loss of safe distances, emergency services and airspace structures) that may cause deviations from standard paths. These results made allowance, both for the clarification of causes of trajectory deviations from nominal paths, and cross-route comparison of the efficiency of air-traffic flows across a range of flight scenarios. Ultimately, the knowledge garnered from trajectory data analytics regarding traffic flows and their performance can provide the basis for developing new air traffic management strategies, procedures and decision-support tools.
2. The traffic flow prediction model will guide the UTM operator regarding upcoming hot spots in an early time horizon. Based on this complexity prediction map, the operator can regulate by reconfiguring the trajectories of flights that would be involved in the hot spots ahead of time in order to avoid their formation. The model can also be fed as an input to mitigate the risks for UTM system, which would suggest actions on trajectories to avoid any conflict (e.g., changing heading, speed, or ground delay) and prevent the formation of congested areas.
3. Deploying an advisory system able to indicate the most appropriate regions for UAM operation and address the trade-off between explicability and performance. In other words, it will generate an intelligent system that will be both explicable and reasonably comprehensible. This will assist the UTM authority, first, via more effective airspace-capacity estimation and, second, in supporting the formulation of new operational regulations and

performance requirements - for instance, making sure there are clear rules for risk management and compliance when AI is used to help keep operations safe in a UTM environment.

7.2 Future Work

During In this study, the proposed framework has been implemented and tested using a simulated dataset that reflects realistic UTM traffic scenarios. However, it is important to acknowledge the necessity for future advancements to further mature these techniques. The insights and findings obtained from this research should serve as motivation to enhance and refine these techniques. Subsequent work should concentrate on ensuring the applicability and effectiveness of these techniques in more complex real-world UTM scenarios. By addressing these considerations, future research can contribute to the development of more mature techniques that effectively optimize airspace management and improve the safety and capacity of UTM systems. During the course of the present research, several areas have been identified for future study, which are outlined below:

1. The proposed framework in this study is implemented using a simulated dataset based on some realistic UTM traffic scenarios. There is scope to extend this further by using actual UTM data when it becomes publicly available.
2. Although the PSO algorithm demonstrates good performance for finding an optimal path that can take realistic scenarios into consideration, the path optimization of the UAVs can be useful in minimizing the flying cost and efficiency of UAVs. A deep learning model integrating reinforcement learning (RL) and a deep neural network (DNN) algorithm are promising candidates for optimizing the path policy in order to maximize the energy efficiency of UAVs [376].
3. In this study, the viability of the flight trajectory data analytics framework for the Bedfordshire UTM airspace area was validated. For future research, one promising area is the deployment of the framework to characterize larger-scale air traffic flows. For instance, characterization of air traffic flows at the

national level might provide valuable information for future UTM traffic management.

4. Since UTM traffic flow patterns identified in this work considered only 2D view of the trajectory data that corresponds to positional information only, it would be beneficial to extend this work to offer a 3D view by taking into account level or altitude information. This would also address more practical issues for UAM operations in 3D.
5. The UTM data-analytics framework comprises generally applicable methods. Nonetheless, there will be a need for certain adaptations, such as clustering-oriented data representation, so that the analysis may be customized for the desired application. For instance, traffic flow pattern predictions could be improved if researchers consider the application of the above work regarding UTM congestion, traffic flow and capacity measurement for UK airspace regions that have a more diverse airspace configurations and expect a much denser UAV traffic flow.
6. Future research in UTM traffic-flow prediction should focus on topics such as improving DL-model efficacy and precision, the selection of more time-efficient learning and training algorithms, network-parameter optimization, and the accommodation of more diverse UTM traffic datasets.
7. It is observed from the literature that most previous studies have predicted traffic flow for short-term horizon only. Costs may be reduced, and airspace better managed, via the long-term prediction horizon. In order to accommodate huge traffic-data volumes for that horizon, suitable big-data approaches must therefore be developed. Consequently, DL-based traffic-flow prediction models should be executed within a big-data paradigm (parallel and/or cloud computing), so that predictions can be achieved easily and relatively cheaply. Within the present field, such initiatives have not hitherto been proposed. Modelling via a non-structured dataset also remains an open challenge in this context.
8. The proposed hybrid recommendation system methodology was developed off-line in MATALB and relies on historical simulated data. An effort may be warranted to create an on-line decision support system that incorporates live

data streams from multiple missions in real time scenarios. The subsequent hardware-related performance time challenges may be dealt using GPUs or cognitive neural chips.

9. Various X-AI methods have begun to generate promising results, but this research field remains in its infancy. Consequently, there are no extant solutions to provide certifiable UTM system support for dynamic capacity management and airspace safety. For these reasons, one potential research direction would involve extending this study's proposed interpretation framework using other forms of trustworthy AI techniques, based on global and local explanation methodologies (SHAP and LIME). This would strengthen trust, mutually, between intelligent systems and humans.

References

[1] M. Intelligence, “DRONES MARKET - GROWTH, TRENDS, COVID-19 IMPACT, AND FORECASTS (2021 - 2026).pdf.” 2020.

[2] Z. Liu, Z. Li, B. Liu, X. Fu, I. Raptis, and K. Ren, “Rise of Mini-Drones: Applications and Issues,” *Proc. Int. Symp. Mob. Ad Hoc Netw. Comput.*, pp. 7–12, 2015, doi: 10.1145/2757302.2757303.

[3] C. Sutheerakul, N. Kronprasert, M. Kaewmoracharoen, and P. Pichayapan, “Application of Unmanned Aerial Vehicles to Pedestrian Traffic Monitoring and Management for Shopping Streets,” *Transp. Res. Procedia*, vol. 25, pp. 1717–1734, 2017, doi: 10.1016/j.trpro.2017.05.131.

[4] A. Restas, “Drone Applications for Supporting Disaster Management,” *World J. Eng. Technol.*, vol. 03, no. 03, pp. 316–321, 2015, doi: 10.4236/wjet.2015.33c047.

[5] J. A. Besada *et al.*, “Drone mission definition and implementation for automated infrastructure inspection using airborne sensors,” *Sensors (Switzerland)*, vol. 18, no. 4, pp. 1–29, 2018, doi: 10.3390/s18041170.

[6] D. L. Wright, V. P. Rasmussen, R. D. Ramsey, D. J. Baker, and J. W. Ellsworth, “Canopy reflectance estimation of wheat nitrogen content for grain protein management,” *GIScience Remote Sens.*, vol. 41, no. 4, pp. 287–300, 2004, doi: 10.2747/1548-1603.41.4.287.

[7] M. J. Duffy, S. Wakayama, R. Hupp, R. Lacy, and M. Stauffer, “A study in reducing the cost of vertical flight with electric propulsion,” *17th AIAA Aviat. Technol. Integr. Oper. Conf. 2017*, no. June, pp. 1–24, 2017, doi: 10.2514/6.2017-3442.

[8] P. Pradeep and P. Wei, “Energy efficient arrival with rta constraint for urban evtol operations,” *AIAA Aerosp. Sci. Meet. 2018*, vol. 0, no. 210059, pp. 1–13, 2018, doi: 10.2514/6.2018-2008.

[9] W. J. Fredericks, M. D. Moore, and R. C. Busan, “Benefits of hybrid-electric

propulsion to achieve 4x increase in cruise efficiency for a VTOL aircraft," *2013 Int. Powered Lift Conf.*, pp. 1–21, 2013, doi: 10.2514/6.2013-4324.

[10] T. Prevot, J. Homola, and J. Mercer, "From rural to urban environments: Human/systems simulation research for low altitude UAS traffic management (UTM)," *16th AIAA Aviat. Technol. Integr. Oper. Conf.*, no. June, pp. 1–12, 2016, doi: 10.2514/6.2016-3291.

[11] P. D. Vascik and R. John Hansman, "Constraint identification in on demand mobility for aviation through an exploratory case study of Los Angeles," *17th AIAA Aviat. Technol. Integr. Oper. Conf. 2017*, no. June 2017, pp. 1–25, 2017, doi: 10.2514/6.2017-3083.

[12] R. Rumba and A. Nikitenko, "The wild west of drones: A review on autonomous- UAV traffic-management," *2020 Int. Conf. Unmanned Aircr. Syst. ICUAS 2020*, pp. 1317–1322, 2020, doi: 10.1109/ICUAS48674.2020.9214031.

[13] P. D. Vascik, H. Balakrishnan, and R. J. Hansman, "Assessment of Air Traffic Control for Urban Air Mobility and Unmanned Systems," *8th Int. Conf. Res. Air Transp.*, no. June, 2018.

[14] M. Radmanesh, M. Kumar, A. Nemati, and M. Sarim, "Dynamic optimal UAV trajectory planning in the National Airspace System via mixed integer linear programming," *Proc. Inst. Mech. Eng. Part G J. Aerosp. Eng.*, vol. 230, no. 9, pp. 1668–1682, 2016, doi: 10.1177/0954410015609361.

[15] FAA, "Concept of Operations v2.0," *Enabling Civ. Low-altitude Airsp. Unmanned Aircr. Syst. Oper.*, p. https://utm.arc.nasa.gov/index.shtml, 2020.

[16] S. J. Undertaking, "U-space blueprint," *Publ. Off. Eur. Union*, pp. 1–12, 2017.

[17] J. A. Pérez-Castán, F. G. Comendador, A. B. Cardenas-Soria, D. Janisch, and R. M. A. Valdés, "Identification, categorisation and gaps of safety indicators for U-space," *Energies*, vol. 13, no. 3, pp. 1–17, 2020, doi:

10.3390/en13030608.

[18] J. Holden and N. Goel, “Fast-Forwarding to a Future of On-Demand Urban Air Transportation,” pp. 1–98, 2016, [Online]. Available: https://www.uber.com/elevate.pdf.

[19] Airbus, “Blueprint for the Sky: the Roadmap for the Safe Integration of Autonomous Aircraft,” 2018.

[20] P. Kopardekar, J. Rios, T. Prevot, M. Johnson, J. Jung, and J. E. Robinson, “Unmanned aircraft system traffic management (UTM) concept of operations,” *16th AIAA Aviat. Technol. Integr. Oper. Conf.*, pp. 1–16, 2016.

[21] E. Mueller, P. Kopardekar, and K. Goodrich, “Enabling airspace integration for high-density on-demand mobility operations,” *17th AIAA Aviat. Technol. Integr. Oper. Conf. 2017*, no. June, pp. 1–24, 2017, doi: 10.2514/6.2017-3086.

[22] R. Shrestha, I. Oh, and S. Kim, “A Survey on Operation Concept, Advancements, and Challenging Issues of Urban Air Traffic Management,” *Front. Futur. Transp.*, vol. 2, no. April, pp. 1–20, 2021, doi: 10.3389/ffutr.2021.626935.

[23] T. Jiang, J. Geller, and J. Collura, “Unmanned Aircraft System traffic management: Concept of operation and system architecture,” *Int. J. Transp. Sci. Technol.*, vol. 5, no. 3, pp. 123–135, 2016, doi: 10.1016/J.IJTST.2017.01.004.

[24] B. Syd Ali, “Traffic management for drones flying in the city,” *Int. J. Crit. Infrastruct. Prot.*, vol. 26, p. 100310, 2019, doi: 10.1016/j.ijcip.2019.100310.

[25] B. Aydin, “Public acceptance of drones: Knowledge, attitudes, and practice,” *Technol. Soc.*, vol. 59, no. August, p. 101180, 2019, doi: 10.1016/j.techsoc.2019.101180.

[26] Icao, “Unmanned Aircraft Systems Traffic Management (UTM) – A

Common Framework with Core Principles for Global Harmonization," *UTM Guid.*, pp. 1–25, 2019, [Online]. Available: https://www.icao.int/safety/UA/Documents/UTM-Framework.en.alltext.pdf.

[27] T. Prevot, J. Rios, P. Kopardekar, J. E. Robinson Iii, M. Johnson, and J. Jung, "UAS Traffic Management (UTM) Concept of Operations to Safely Enable Low Altitude Flight Operations," no. June, 2016, doi: 10.2514/6.2016-3292.

[28] S. H. Chung, H. L. Ma, M. Hansen, and T. M. Choi, "Data science and analytics in aviation," *Transp. Res. Part E Logist. Transp. Rev.*, vol. 134, no. January, 2020, doi: 10.1016/j.tre.2020.101837.

[29] R. Conde, D. Alejo, J. A. Cobano, A. Viguria, and A. Ollero, "Conflict detection and resolution method for cooperating unmanned aerial vehicles," *J. Intell. Robot. Syst. Theory Appl.*, vol. 65, no. 1–4, pp. 495–505, 2012, doi: 10.1007/s10846-011-9564-6.

[30] R. Alligier, D. Gianazza, and N. Durand, "Machine Learning and Mass Estimation Methods for Ground-Based Aircraft Climb Prediction," *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 6, pp. 3138–3149, 2015, doi: 10.1109/TITS.2015.2437452.

[31] N. Takeichi, R. Kaida, A. Shimomura, and T. Yamauchi, "Prediction of delay due to air traffic control by machine learning," *AIAA Model. Simul. Technol. Conf. 2017*, no. January, pp. 1–8, 2017, doi: 10.2514/6.2017-1323.

[32] H. Lee, "Taxi-out Time Prediction for Departures at Charlotte Airport," 2019.

[33] S. Ayhan and H. Samet, "Aircraft trajectory prediction made easy with predictive analytics," *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, vol. 13-17-Augu, pp. 21–30, 2016, doi: 10.1145/2939672.2939694.

[34] Z. Sándor, "Challenges caused by the unmanned aerial vehicle in the air traffic management," *Period. Polytech. Transp. Eng.*, vol. 47, no. 2, pp. 96–105, 2019, doi: 10.3311/PPtr.11204.

[35] S. Bijjahalli, R. Sabatini, and A. Gardi, “Advances in intelligent and autonomous navigation systems for small UAS,” *Prog. Aerosp. Sci.*, vol. 115, no. June, p. 100617, 2020, doi: 10.1016/j.paerosci.2020.100617.

[36] Z. Lv, H. Song, P. Basanta-Val, A. Steed, and M. Jo, “Next-Generation Big Data Analytics: State of the Art, Challenges, and Future Research Topics,” *IEEE Trans. Ind. Informatics*, vol. 13, no. 4, pp. 1891–1899, 2017, doi: 10.1109/TII.2017.2650204.

[37] B. Wang, V. Kharchenko, A. Kukush, and N. Kuzmenko, “Unmanned aerial vehicles trajectory analysis considering missing data,” *Transport*, vol. 34, no. 2, pp. 155–162, 2019, doi: 10.3846/transport.2019.8544.

[38] S. Zsolt, “Unmanned Peer-Reviewed Article Model of the System of Information for the Traffic Management of Unmanned,” *J. UNMANNED Aer. Syst. Introd.*, vol. 4, no. 1, pp. 18–31, 2019.

[39] A. Majumdar, W. Y. Ochieng, J. Bentham, and M. Richards, “En-route sector capacity estimation methodologies: An international survey,” *J. Air Transp. Manag.*, vol. 11, no. 6, pp. 375–387, 2005, doi: 10.1016/j.jairtraman.2005.05.002.

[40] A. Klein, L. Cook, B. Wood, and D. Simenauer, “Airspace Capacity Estimation Using Flows And Weather-Impacted Traffic Index The Task of Translating Weather Information into TFM Constraints Scope of Current Research.”

[41] E. Sunil *et al.*, “How Do Layered Airspace Design Parameters Affect Airspace Capacity and Safety ? The Influence of Traffic Structure on Airspace Capacity To cite this version : HAL Id : hal-02967017,” 2020.

[42] M. N. CİZRELİOĞULLARI, P. BARUT, and T. IMANOV, “Future Air Transportation Ramification: Urban Air Mobility (Uam) Concept: Urban Air Mobility,” *Prizren Soc. Sci. J.*, vol. 6, no. 2, pp. 24–31, 2022, doi: 10.32936/pssj.v6i2.335.

[43] I. Khoufi, A. Laouiti, C. Adjih, and M. Hadded, “UAVs Trajectory

Optimization for Data Pick Up and Delivery with Time Window," *Drones*, vol. 5, no. 2, pp. 1–22, 2021, doi: 10.3390/drones5020027.

[44] A. M. Raivi, S. M. A. Huda, M. M. Alam, and S. Moh, "Drone Routing for Drone-Based Delivery Systems: A Review of Trajectory Planning, Charging, and Security," *Sensors*, vol. 23, no. 3, 2023, doi: 10.3390/s23031463.

[45] Civil Aviation Authority, "Beyond Visual Line of Sight in Non-Segregated Airspace. CAP 1861," no. October, pp. 1–11, 2020, [Online]. Available: https://publicapps.caa.co.uk/docs/33/CAP 1861 - BVLOS Fundamentals v2.pdf.

[46] P. Kopardekar, T. Prevot, J. Rios, J. E. Robinson III, J. Jung, and M. Johnson, "UAS Traffic Management (UTM) Concept of Operations to Safely Enable Low Altitude Flight Operations," no. June, pp. 1–16, 2016, doi: 10.2514/6.2016-3292.

[47] A. Hately *et al.*, "U-Space Concept of Operations," *Corus*, vol. v.3.0, no. October 2019, pp. 1–92, 2019.

[48] L. Legros, R. Garrity, and A. Hately, "Initial view on Principles for the U-space architecture," pp. 1–19, 2019, [Online]. Available: https://www.sesarju.eu/sites/default/files/documents/u-space/SESAR principles for U-space architecture.pdf.

[49] C. Huang, Z. Ming, and H. Huang, "Drone Stations-Aided Beyond-Battery-Lifetime Flight Planning for Parcel Delivery," *IEEE Trans. Autom. Sci. Eng.*, pp. 1–11, 2022, doi: 10.1109/TASE.2022.3213254.

[50] S. Aggarwal and N. Kumar, "Path planning techniques for unmanned aerial vehicles: A review, solutions, and challenges," *Comput. Commun.*, vol. 149, no. July 2019, pp. 270–299, 2020, doi: 10.1016/j.comcom.2019.10.014.

[51] S. M. A. Huda and S. Moh, "Survey on computation offloading in UAV-Enabled mobile edge computing," *J. Netw. Comput. Appl.*, vol. 201, no.

January, p. 103341, 2022, doi: 10.1016/j.jnca.2022.103341.

[52] M. M. Alam, M. Y. Arafat, S. Moh, and J. Shen, “Topology control algorithms in multi-unmanned aerial vehicle networks: An extensive survey,” *J. Netw. Comput. Appl.*, vol. 207, no. March, p. 103495, 2022, doi: 10.1016/j.jnca.2022.103495.

[53] M. Y. Arafat and S. Moh, “JRCS: Joint Routing and Charging Strategy for Logistics Drones,” *IEEE INTERNET THINGS J.*, vol. 9, no. 21, 2022, doi: 10.1109/JIOT.2022.3182750.

[54] K. Dorling, J. Heinrichs, G. G. Messier, and S. Magierowski, “Vehicle Routing Problems for Drone Delivery,” *IEEE Trans. Syst. Man, Cybern. Syst.*, vol. 47, no. 1, pp. 70–85, 2017, doi: 10.1109/TSMC.2016.2582745.

[55] J. Homola, N. Craven, P. Verma, and V. Baskaran, “Strategic Deconfliction Performance Results and Analysis from the NASA UTM Technical Capability Level 4 Demonstration,” no. August, pp. 1–26, 2020.

[56] “CORUS Project U-Space Concept of Operations (U-Space ConOps). Available online: https://www.eurocontrol.int/project/ concept-operations-european-utm-systems.” .

[57] K. W. Chen, M. R. Xie, Y. M. Chen, T. T. Chu, and Y. B. Lin, “DroneTalk: An Internet-of-Things-Based Drone System for Last-Mile Drone Delivery,” *IEEE Trans. Intell. Transp. Syst.*, vol. 23, no. 9, pp. 15204–15217, Sep. 2022, doi: 10.1109/TITS.2021.3138432.

[58] F. B. Sorbelli, F. Corò, S. K. Das, and C. M. Pinotti, “Energy-Constrained Delivery of Goods With Drones Under Varying Wind Conditions,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, 2021, doi: 10.1109/TITS.2020.3044420.

[59] A. Hamdi, F. D. Salim, D. Y. Kim, A. G. Neiat, and A. Bouguettaya, “Drone-as-a-Service Composition Under Uncertainty,” *IEEE Trans. Serv. Comput.*, vol. 15, no. 5, pp. 2685–2698, 2022, doi: 10.1109/TSC.2021.3066006.

[60] S. Sawadsitang, D. Niyato, P. S. Tan, P. Wang, and S. Nutanong, “Shipper Cooperation in Stochastic Drone Delivery: A Dynamic Bayesian Game Approach,” *IEEE Trans. Veh. Technol.*, vol. 70, no. 8, pp. 7437–7452, 2021, doi: 10.1109/TVT.2021.3090992.

[61] O. International Civil Aviation, *Doc. 4444 - Procedures for Air Navigation Services - Air Traffic Management*. 2016.

[62] D. Alejo, R. Conde, J. A. Cabana, and A. Ollero, “Multi-UAV collision avoidance with separation assurance under uncertainties,” *IEEE 2009 Int. Conf. Mechatronics, ICM 2009*, vol. 00, no. April, pp. 1–6, 2009, doi: 10.1109/ICMECH.2009.4957235.

[63] B. M. and N. A., “A Conceptual Framework and a Review of Conflict Sensing, Detection, Awareness and Escape Maneuvering Methods for UAVs,” *Aeronaut. Astronaut.*, no. May 2014, 2011, doi: 10.5772/26567.

[64] E. Calvo-Fernández, L. Perez-Sanz, J. M. Cordero-García, and R. M. Arnaldo-Valdés, “Conflict-free trajectory planning based on a data-driven conflict-resolution model,” *J. Guid. Control. Dyn.*, vol. 40, no. 3, pp. 615–627, 2017, doi: 10.2514/1.G000691.

[65] A. Evans, M. Egorov, and S. Munn, “Fairness in decentralized strategic deconfliction in utm,” *AIAA Scitech 2020 Forum*, vol. 1 PartF, no. January, pp. 1–16, 2020, doi: 10.2514/6.2020-2203.

[66] L. Wang, X. Peng, F. Xu, J. Xing, Z. Zhou, and X. Tian, “Improved Binary Space Partition Method of Airspace Sectorization Considering Multiple Scenarios,” *Proc. 2022 IEEE 4th Int. Conf. Civ. Aviat. Saf. Inf. Technol. ICCASIT 2022*, pp. 456–460, 2022, doi: 10.1109/ICCASIT55263.2022.9986566.

[67] N. Pongsakornsathien *et al.*, “A performance-based airspace model for unmanned aircraft systems traffic management,” *Aerospace*, vol. 7, no. 11, pp. 1–25, 2020, doi: 10.3390/aerospace7110154.

[68] Y. Tang, Y. Xu, and G. Inalhan, “An Integrated Approach for On-Demand

Dynamic Capacity Management Service in U-Space," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 5, pp. 4180–4195, 2022, doi: 10.1109/TAES.2022.3159317.

[69] E. Tijan, "Utilization of Aerial Drone Technology in Logistics," vol. 63, no. April, pp. 27–37, 2023.

[70] B. Sah, R. Gupta, and D. Bani-Hani, "Analysis of barriers to implement drone logistics," *Int. J. Logist. Res. Appl.*, vol. 24, no. 6, pp. 531–550, 2021, doi: 10.1080/13675567.2020.1782862.

[71] S. Çıkmak, G. Kırbaç, and B. Kesici, "Analyzing the Challenges to Adoption of Drones in the Logistics Sector Using the Best-Worst Method," *Bus. Econ. Res. J.*, vol. 14, no. 2, pp. 227–242, 2023, doi: 10.20409/berj.2023.413.

[72] T. McCarthy, L. Pforte, and R. Burke, "Fundamental elements of an urban UTM," *Aerospace*, vol. 7, no. 7, 2020, doi: 10.3390/AEROSPACE7070085.

[73] S. Ahmad and V. Saxena, "Design of formal air traffic control system through UML," *Ubiquitous Comput. Commun. J.*, vol. 3, no. 6, pp. 11–20, 2008.

[74] P. Janik, M. Zawistowski, R. Fellner, and G. Zawistowski, "Unmanned aircraft systems risk assessment based on sora for first responders and disaster management," *Appl. Sci.*, vol. 11, no. 12, 2021, doi: 10.3390/app11125364.

[75] H. Habibi, D. M. K. K. Venkateswara Rao, J. Luis Sanchez-Lopez, and H. Voos, "On SORA for High-Risk UAV Operations under New EU Regulations: Perspectives for Automated Approach," *2023 Int. Conf. Unmanned Aircr. Syst. ICUAS 2023*, pp. 213–220, 2023, doi: 10.1109/ICUAS57906.2023.10156517.

[76] "Easy Access Rules for Unmanned Aircraft Systems - Revision from September 2022 | EASA." https://www.easa.europa.eu/en/document-library/easy-access-rules/online-publications/easy-access-rules-unmanned-aircraft-systems (accessed Oct. 25, 2023).

[77] ICAO, “ICAO Model UAS Regulations,” *2021*. https://www.icao.int/safety/UA/Pages/ICAO-Model-UAS-Regulations.aspx (accessed Aug. 09, 2023).

[78] A. Bauranov and J. Rakas, “Designing airspace for urban air mobility: A review of concepts and approaches,” *Prog. Aerosp. Sci.*, vol. 125, p. 100726, 2021, doi: 10.1016/j.paerosci.2021.100726.

[79] T. A. Hearn, M. T. Kotwicz Herniczek, and B. J. German, “Conceptual Framework for Dynamic Optimal Airspace Configuration for Urban Air Mobility,” *J. Air Transp.*, vol. 31, no. 2, pp. 68–82, 2023, doi: 10.2514/1.D0327.

[80] A. Alharbi, I. Petrunin, and D. Panagiotakopoulos, “Identification and Characterization of Traffic Flow Patterns for UTM application,” *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, vol. 2021-Octob, 2021, doi: 10.1109/DASC52595.2021.9594494.

[81] E. Ebeid, M. Skriver, K. H. Terkildsen, K. Jensen, and U. P. Schultz, “A survey of Open-Source UAV flight controllers and flight simulators,” *Microprocess. Microsyst.*, vol. 61, no. January 2020, pp. 11–20, 2018, doi: 10.1016/j.micpro.2018.05.002.

[82] S. Bijjahalli, R. Sabatini, and A. Gardi, “Advances in intelligent and autonomous navigation systems for small UAS,” *Prog. Aerosp. Sci.*, vol. 115, no. December 2019, p. 100617, 2020, doi: 10.1016/j.paerosci.2020.100617.

[83] S. Yoon, D. Shin, Y. Choi, and K. Park, “Development of a Flexible and Expandable UTM Simulator Based on Open Sources and Platforms,” *Aerospace*, vol. 8, no. 5, p. 133, 2021, doi: 10.3390/aerospace8050133.

[84] S. Razzaq, C. Xydeas, M. E. Everett, A. Mahmood, and T. Alquthami, “Three-Dimensional UAV Routing with Deconfliction,” *IEEE Access*, vol. 6, no. April, pp. 21536–21551, 2018, doi: 10.1109/ACCESS.2018.2824558.

[85] A. Al-Mousa, B. H. Sababha, N. Al-Madi, A. Barghouthi, and R. Younisse,

"UTSim: A framework and simulator for UAV air traffic integration, control, and communication," *Int. J. Adv. Robot. Syst.*, vol. 16, no. 5, pp. 1–19, 2019, doi: 10.1177/1729881419870937.

[86] J. Homola, T. Prevot, J. Mercer, N. Bienert, and C. Gabriel, "UAS traffic management (UTM) simulation capabilities and laboratory environment," *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, vol. 2016-Decem, pp. 1–7, 2016, doi: 10.1109/DASC.2016.7778078.

[87] Z. Zhao *et al.*, "A simulation framework for fast design space exploration of unmanned air system traffic management policies," *arXiv*, 2019.

[88] J. A. Millan-Romera, J. J. Acevedo, A. R. Castano, H. Perez-Leon, C. Capitan, and A. Ollero, "A UTM simulator based on ROS and Gazebo," *2019 Int. Work. Res. Educ. Dev. Unmanned Aer. Syst. RED-UAS 2019*, pp. 132–141, 2019, doi: 10.1109/REDUAS47371.2019.8999705.

[89] "UNIFLY: Software - Simulation - AeroExpo." https://www.aeroexpo.online/prod/unifly-185506.html (accessed Aug. 28, 2022).

[90] "AirMap | AirMap - Unmanned traffic management software solutions |…." https://www.airmap.com/ (accessed Aug. 28, 2022).

[91] L. Sedov and V. Polishchuk, "Centralized and Distributed UTM in Layered Airspace," pp. 1–8, 2018.

[92] F. Ho, R. Geraldes, A. Gonc, M. Cavazza, and H. Prendinger, "Improved Conflict Detection and Resolution for Service UAVs in Shared Airspace," *IEEE Trans. Veh. Technol.*, vol. PP, p. 1, doi: 10.1109/TVT.2018.2889459.

[93] D. Camacho, "Design and Development of a Lightweight Multi-UAV Simulator."

[94] S. Shah, D. Dey, C. Lovett, and A. Kapoor, "AirSim: High-Fidelity Visual and physical simulation for autonomous vehicles," *arXiv*, pp. 1–14, 2017, doi: 10.1007/978-3-319-67361-5_40.

[95] N. Koenig and A. Howard, “Design and Use Paradigms for Gazebo , An Open-Source Multi-Robot Simulator,” 2004.

[96] J. A. Millan-Romera, J. J. Acevedo, Á. R. Castaño, H. Perez-Leon, C. Capitán, and A. Ollero, “A UTM simulator based on ROS and Gazebo,” in *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*, 2019, pp. 132–141.

[97] “Unity Real-Time Development Platform | 3D, 2D VR & AR Engine.” https://unity.com/ (accessed Aug. 28, 2022).

[98] J. M. Hoekstra and J. Ellerbroek, “BlueSky ATC simulator project: an open-data and open-source approach,” *Proc. 7th Int. Conf. Res. Air Transp.*, pp. 1–8, 2016.

[99] J. Nguyenvu, P. Nguyen, and M. Abdulrahim, “Development of Unmanned Traffic Management Simulations with Monte Carlo and Unreal Engine,” *AIAA Sci. Technol. Forum Expo. AIAA SciTech Forum 2022*, pp. 1–12, 2022, doi: 10.2514/6.2022-1918.

[100] J. A. Millan-Romera, J. J. Acevedo, Á. R. Castaño, H. Perez-Leon, C. Capitán, and A. Ollero, “A UTM simulator based on ROS and Gazebo,” in *2019 Workshop on Research, Education and Development of Unmanned Aerial Systems (RED UAS)*, 2019, pp. 132–141.

[101] “OpnScope Air Traffic Control Simulator,” 2019. https://www.openscope.co/.

[102] E. Ebeid, M. Skriver, K. H. Terkildsen, K. Jensen, and U. P. Schultz, “A survey of open-source UAV flight controllers and flight simulators,” *Microprocess. Microsyst.*, vol. 61, pp. 11–20, 2018.

[103] P. Boccadoro and A. Cardellicchio, “PLANE: An Extensible open source framework for modeling the internet of drones,” *arXiv*, 2019.

[104] J. Giesbrecht, B. Touchton, T. Galluzzo, D. Kent, C. D. Crane, and J. Giesbrecht, “Global Path Planning for Unmanned Ground Vehicles,” *From*

OAIster, Provid. by OCLC Coop., no. December, pp. 1–41, 2004.

[105] Y. S. Jiao, X. M. Wang, H. Chen, and Y. Li, “Research on the coverage path planning of UAVs for polygon areas,” *Proc. 2010 5th IEEE Conf. Ind. Electron. Appl. ICIEA 2010*, pp. 1467–1472, 2010, doi: 10.1109/ICIEA.2010.5514816.

[106] Y. Li, H. Chen, M. Joo Er, and X. Wang, “Coverage path planning for UAVs based on enhanced exact cellular decomposition method,” *Mechatronics*, vol. 21, no. 5, pp. 876–885, 2011, doi: 10.1016/j.mechatronics.2010.10.009.

[107] M. Torres, D. A. Pelta, J. L. Verdegay, and J. C. Torres, “Coverage path planning with unmanned aerial vehicles for 3D terrain reconstruction,” *Expert Syst. Appl.*, vol. 55, pp. 441–451, 2016, doi: 10.1016/j.eswa.2016.02.007.

[108] A. Xu, C. Viriyasuthee, and I. Rekleitis, “Optimal complete terrain coverage using an unmanned aerial vehicle,” *Proc. - IEEE Int. Conf. Robot. Autom.*, pp. 2513–2519, 2011, doi: 10.1109/ICRA.2011.5979707.

[109] J. J. Acevedo, B. C. Arrue, I. Maza, and A. Ollero, “Cooperative large area surveillance with a team of aerial mobile robots for long endurance missions,” *J. Intell. Robot. Syst. Theory Appl.*, vol. 70, no. 1–4, pp. 329–345, 2013, doi: 10.1007/s10846-012-9716-3.

[110] I. Koohi and V. Z. Groza, “Optimizing Particle Swarm Optimization algorithm,” *Can. Conf. Electr. Comput. Eng.*, pp. 1–5, 2014, doi: 10.1109/CCECE.2014.6901057.

[111] H. P. Dai, D. D. Chen, and Z. S. Zheng, “Effects of random values for particle swarm optimization algorithm,” *Algorithms*, vol. 11, no. 2, pp. 1–20, 2018, doi: 10.3390/A11020023.

[112] Basavanna. M, “An Overview of Path Planning and Obstacle Avoidance Algorithms in Mobile Robots,” *Int. J. Eng. Res.*, vol. V8, no. 12, pp. 478–482, 2019, doi: 10.17577/ijertv8is120252.

[113] K. L. Lim, L. S. Yeong, S. I. Ch'Ng, K. P. Seng, and L. M. Ang, "Uninformed multigoal pathfinding on grid maps," *Proc. - 2014 Int. Conf. Inf. Sci. Electron. Electr. Eng. ISEEE 2014*, vol. 3, pp. 1552–1556, 2014, doi: 10.1109/InfoSEEE.2014.6946181.

[114] D. S. Jang, H. J. Chae, and H. L. Choi, "Optimal control-based UAV path planning with dynamically-constrained TSP with neighborhoods," *Int. Conf. Control. Autom. Syst.*, vol. 2017-Octob, no. Iccas, pp. 373–378, 2017, doi: 10.23919/ICCAS.2017.8204468.

[115] A. Budiyanto, A. Cahyadi, T. B. Adji, and O. Wahyunggoro, "UAV obstacle avoidance using potential field under dynamic environment," *ICCEREC 2015 - Int. Conf. Control. Electron. Renew. Energy Commun.*, pp. 187–192, 2015, doi: 10.1109/ICCEREC.2015.7337041.

[116] J. Dai, Y. Wang, C. Wang, J. Ying, and J. Zhai, "Research on Hierarchical Potential Field Method of Path Planning for UAVs," *Proc. 2018 2nd IEEE Adv. Inf. Manag. Commun. Electron. Autom. Control Conf. IMCEC 2018*, no. Imcec, pp. 529–535, 2018, doi: 10.1109/IMCEC.2018.8469312.

[117] W. Bai *et al.*, "A Cooperative Route Planning Method for Multi-UAVs Based-on the Fusion of Artificial Potential Field and B-spline Interpolation," *Chinese Control Conf. CCC*, vol. 2018-July, pp. 6733–6738, 2018, doi: 10.23919/ChiCC.2018.8483665.

[118] T. T. Mac, C. Copot, A. Hernandez, and R. De Keyser, "Improved potential field method for unknown obstacle avoidance using UAV in indoor environment," *SAMI 2016 - IEEE 14th Int. Symp. Appl. Mach. Intell. Informatics - Proc.*, pp. 345–350, 2016, doi: 10.1109/SAMI.2016.7423032.

[119] H. V. Abeywickrama, B. A. Jayawickrama, Y. He, and E. Dutkiewicz, "Potential field based inter-UAV collision avoidance using virtual target relocation," *IEEE Veh. Technol. Conf.*, vol. 2018-June, pp. 1–5, 2018, doi: 10.1109/VTCSpring.2018.8417773.

[120] M. Radmanesh, M. Kumar, P. H. Guentert, and M. Sarim, "Overview of

Path-Planning and Obstacle Avoidance Algorithms for UAVs: A Comparative Study," *Unmanned Syst.*, vol. 6, no. 2, pp. 95–118, 2018, doi: 10.1142/S2301385018400022.

[121] A. Sathyan, N. Boone, and K. Cohen, "Comparison of Approximate Approaches to Solving the Travelling Salesman Problem and its Application to UAV Swarming," *Int. J. Unmanned Syst. Eng.*, vol. 3, no. 1, pp. 1–16, 2015, doi: 10.14323/ijuseng.2015.1.

[122] M. Hafidi, M. Benaddy, and S. Krit, "Review of optimization and automation of air traffic control systems," vol. 7, no. 3, pp. 1–7, 2018, doi: 10.1145/3234698.3234708.

[123] Z. Abdmouleh, A. Gastli, L. Ben-Brahim, M. Haouari, and N. A. Al-Emadi, "Review of optimization techniques applied for the integration of distributed generation from renewable energy sources," *Renew. Energy*, vol. 113, no. May, pp. 266–280, 2017, doi: 10.1016/j.renene.2017.05.087.

[124] I. Rahman, P. M. Vasant, B. S. Mahinder Singh, and M. Abdullah-Al-Wadud, "Swarm Intelligence-Based Smart Energy Allocation Strategy for Charging Stations of Plug-In Hybrid Electric Vehicles," *Math. Probl. Eng.*, vol. 2015, no. December 2014, 2015, doi: 10.1155/2015/620425.

[125] M. Radmanesh, M. Kumar, P. H. Guentert, and M. Sarim, "Overview of Path-Planning and Obstacle Avoidance Algorithms for UAVs: A Comparative Study," *Unmanned Syst.*, vol. 6, no. 2, pp. 95–118, 2018, doi: 10.1142/S2301385018400022.

[126] H. Liu, X. Li, M. Fan, G. Wu, W. Pedrycz, and P. N. Suganthan, "An Autonomous Path Planning Method for Unmanned Aerial Vehicle based on A Tangent Intersection and Target Guidance Strategy," pp. 1–12.

[127] F. D. Wihartiko, H. Wijayanti, and F. Virgantari, "Performance comparison of genetic algorithms and particle swarm optimization for model integer programming bus timetabling problem," *IOP Conf. Ser. Mater. Sci. Eng.*, vol. 332, no. 1, 2018, doi: 10.1088/1757-899X/332/1/012020.

[128] ICAO, “Doc 9854, Global Air Traffic Management Operational Concept,” *Int. Civ. Aviat. Organ.*, p. 82, 2005, doi: 10.1523/JNEUROSCI.2228-11.2013.

[129] H. Hildmann and E. Kovacs, “Review: Using Unmanned Aerial Vehicles (UAVs) as Mobile Sensing Platforms (MSPs) for Disaster Response, Civil Security and Public Safety,” *Drones*, vol. 3, no. 3, p. 59, 2019, doi: 10.3390/drones3030059.

[130] 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.

[131] L. Sedov, V. Polishchuk, and V. Bulusu, “Sampling-based capacity estimation for unmanned traffic management,” *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, vol. 2017-Septe, 2017, doi: 10.1109/DASC.2017.8101995.

[132] 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, 2000, doi: 10.1109/6979.898217.

[133] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, “Review of conflict resolution methods for manned and unmanned aviation,” *Aerospace*, vol. 7, no. 6, 2020, doi: 10.3390/AEROSPACE7060079.

[134] P. Kopardekar, K. Bilimoria, and B. Sridhar, “Initial concepts for dynamic airspace configuration,” *Collect. Tech. Pap. - 7th AIAA Aviat. Technol. Integr. Oper. Conf.*, vol. 1, no. February 2015, pp. 695–706, 2007, doi: 10.2514/6.2007-7763.

[135] Y. I. Jenie, E. J. Van Kampen, C. C. De Visser, J. Ellerbroek, and J. M. Hoekstra, “Selective velocity obstacle method for deconflicting maneuvers applied to unmanned aerial vehicles,” *J. Guid. Control. Dyn.*, vol. 38, no. 6, pp. 1140–1145, 2015, doi: 10.2514/1.G000737.

[136] A. Salta, R. Geraldes, A. Goncalves, and M. Cavazza, “Multi-Agent Path

Finding for UAV Traffic Management,” pp. 131–139, 2019.

[137] F. Ho, R. Geraldes, A. Goncalves, M. Cavazza, and H. Prendinger, “Improved Conflict Detection and Resolution for Service UAVs in Shared Airspace,” *IEEE Trans. Veh. Technol.*, vol. 68, no. 2, pp. 1231–1242, 2019, doi: 10.1109/TVT.2018.2889459.

[138] Y. I. Jenie, E. J. Van Kampen, J. Ellerbroek, and J. M. Hoekstra, “Taxonomy of Conflict Detection and Resolution Approaches for Unmanned Aerial Vehicle in an Integrated Airspace,” *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 3, pp. 558–567, 2017, doi: 10.1109/TITS.2016.2580219.

[139] D. Šišlák, P. Volf, Š. Kopriva, and M. Pěchouček, “Agentfly: NAS-wide simulation framework integrating algorithms for automated collision avoidance,” *ICNS 2011 - Integr. Commun. Navig. Surveill. Conf. Renov. Glob. Air Transp. Syst. Proc.*, pp. 1–11, 2011, doi: 10.1109/ICNSURV.2011.5935278.

[140] S. Roelofsen, D. Gillet, and A. Martinoli, “Collision avoidance with limited field of view sensing: A velocity obstacle approach,” *Proc. - IEEE Int. Conf. Robot. Autom.*, no. 2, pp. 1922–1927, 2017, doi: 10.1109/ICRA.2017.7989223.

[141] N. Barnier and C. Allignol, “Combining flight level allocation with ground holding to optimize 4D-deconfliction,” *Proc. 9th USA/Europe Air Traffic Manag. Res. Dev. Semin. ATM 2011*, pp. 636–645, 2011.

[142] S. Chaimatanan, D. Delahaye, M. Mongeau, S. Chaimatanan, D. Delahaye, and M. Mongeau, “Strategic deconfliction of aircraft trajectories To cite this version : Strategic deconfliction of aircraft trajectories,” 2013.

[143] V. Duchamp, B. Josefsson, T. Polishchuk, V. Polishchuk, and R. Wiren, “Air Traffic Deconfliction Using Sum Coloring,” 2019.

[144] N. Barnier, C. Allignol, N. Barnier, and C. Allignol, “adjustment To cite this version : 4D-Trajectory Deconfliction Through Departure Time Adjustment,”

2014.

[145] A. D. Evans, M. Egorov, and S. Munn, “Correction: Fairness in Decentralized Strategic Deconfliction in UTM,” pp. 1–17, 2020, doi: 10.2514/6.2020-2203.c1.

[146] ICAO, “Unmanned Aircraft Systems Traffic Management (UTM) – A Common Framework with Core Principles for Global Harmonization,” *UTM Guid.*, pp. 1–25, 2019, [Online]. Available: https://www.icao.int/safety/UA/Documents/UTM-Framework.en.alltext.pdf.

[147] P. Šul’Aj, R. Haluska, L. Ovsenik, S. Marchevsky, P. Pulli, and V. Kramar, “UAV management system for the smart city,” *DISA 2018 - IEEE World Symp. Digit. Intell. Syst. Mach. Proc.*, pp. 119–124, 2018, doi: 10.1109/DISA.2018.8490535.

[148] A. Carrio, C. Sampedro, A. Rodriguez-Ramos, and P. Campoy, “A review of deep learning methods and applications for unmanned aerial vehicles,” *J. Sensors*, vol. 2017, 2017, doi: 10.1155/2017/3296874.

[149] “Enabling Civilian Low-Altitude Airspace and Unmanned Aerial System (UAS) Operations By,” pp. 1–18, 2014.

[150] M. Cond, R. Murça, M. X. Guterres, M. De Oliveira, T. Szenczuk, and W. Silva, “Journal of Air Transport Management Characterizing the Brazilian airspace structure and air traffic performance via trajectory data analytics,” vol. 85, no. November 2019, 2020, doi: 10.1016/j.jairtraman.2020.101798.

[151] X. Oliver, L. Basora, B. Viry, R. Alligier, X. Olive, and B. Viry, “Deep Trajectory Clustering with Autoencoders,” 2020, [Online]. Available: https://hal-enac.archives-ouvertes.fr/hal-02916241.

[152] X. Olive and L. Basora, “Detection and identification of significant events in historical aircraft trajectory data,” *Transp. Res. Part C Emerg. Technol.*, vol. 119, no. December, 2020, doi: 10.1016/j.trc.2020.102737.

[153] M. A. Amaro Carmona, F. Saez Nieto, and C. E. Verdonk Gallego, “A data-

driven methodology for characterization of a terminal manoeuvring area in multi-airport systems," *Transp. Res. Part C Emerg. Technol.*, vol. 111, no. November 2018, pp. 185–209, 2020, doi: 10.1016/j.trc.2019.12.011.

[154] M. C. R. Murça, R. J. Hansman, L. Li, and P. Ren, "Flight trajectory data analytics for characterization of air traffic flows: A comparative analysis of terminal area operations between New York, Hong Kong and Sao Paulo," *Transp. Res. Part C Emerg. Technol.*, vol. 97, no. October, pp. 324–347, 2018, doi: 10.1016/j.trc.2018.10.021.

[155] R. . Condé Rocha Murça, M.; Hansman, "Data-Driven Modeling of Air Traffic Flows for Advanced Air Traffic Management," *Ph.D. Thesis, Massachusetts Inst. Technol. Cambridge, MA, USA*, 2018.

[156] L. Basora, J. Morio, and C. Mailhot, "A trajectory clustering framework to analyse air traffic flows," *SESAR Innov. Days*, no. November, 2017.

[157] X. Olive and J. Morio, "Trajectory clustering of air traffic flows around airports," *Aerosp. Sci. Technol.*, vol. 84, pp. 776–781, 2019, doi: 10.1016/j.ast.2018.11.031.

[158] M. C. R. Murça, M. X. Guterres, M. de Oliveira, J. B. Tarelho Szenczuk, and W. S. S. anna Souza, "Characterizing the Brazilian airspace structure and air traffic performance via trajectory data analytics," *J. Air Transp. Manag.*, vol. 85, no. November 2019, 2020, doi: 10.1016/j.jairtraman.2020.101798.

[159] Z. Feng and Y. Zhu, "A Survey on Trajectory Data Mining: Techniques and Applications," *IEEE Access*, vol. 4, pp. 2056–2067, 2016, doi: 10.1109/ACCESS.2016.2553681.

[160] D. Wang, T. Miwa, and T. Morikawa, *Big Trajectory Data Mining Applications , and Services*. 2020.

[161] Y. Zheng, "Trajectory data mining: An overview," *ACM Trans. Intell. Syst. Technol.*, vol. 6, no. 3, pp. 1–41, 2015, doi: 10.1145/2743025.

[162] G. Antonini, J. P. Thiran, and S. Member, “Using Trajectory Clustering,” vol. 16, no. 8, pp. 1008–1020, 2006.

[163] J. Ackley, T. G. Puranik, and D. N. Mavris, “A supervised learning approach for safety event precursor identification in commercial aviation,” *Aiaa Aviat. 2020 Forum*, vol. 1 PartF, no. June, 2020, doi: 10.2514/6.2020-2880.

[164] D. R. De Almeida, C. D. S. Baptista, and F. G. De Andrade, “A Survey on Big Data for Trajectory Analytics,” pp. 1–24, 2020, doi: 10.3390/ijgi9020088.

[165] C. Parent *et al.*, “Semantic trajectories modeling and analysis,” *ACM Comput. Surv.*, vol. 45, no. 4, 2013, doi: 10.1145/2501654.2501656.

[166] X. Kong *et al.*, “Big trajectory data: A survey of applications and services,” *IEEE Access*, vol. 6, no. October, pp. 58295–58306, 2018, doi: 10.1109/ACCESS.2018.2873779.

[167] J. Bian, D. Tian, Y. Tang, and D. Tao, “A survey on trajectory clustering analysis,” 2018, [Online]. Available: http://arxiv.org/abs/1802.06971.

[168] R. Xu, S. Member, and D. W. Ii, “Survey of Clustering Algorithms,” vol. 16, no. 3, pp. 645–678, 2005.

[169] S. Gaffney, “Trajectory Clustering with Mixtures of Regression Models,” no. July 1999, 2013, doi: 10.1145/312129.312198.

[170] J. Alon, S. Sclaroff, and G. Kollios, “Discovering Clusters in Motion Time-Series Data,” 2003.

[171] O. Maimon and L. Rokach, *Data Mining and Knowledge Discovery Handbook , 2nd ed*, no. January 2010. 2018.

[172] M. Ankerst, M. M. Breunig, H. Kriegel, and J. Sander, “OPTICS: ordering points to identify the clustering structure,” *ACM SIGMOD Rec.*, vol. 28, no. 2, pp. 49–60, 1999.

[173] M. Daszykowski and B. Walczak, “Density-Based Clustering Methods,” *Compr. Chemom.*, vol. 2, pp. 635–654, 2009, doi: 10.1016/B978-

044452701-1.00067-3.

[174] M. Nanni and D. Pedreschi, “Time-focused clustering of trajectories of moving objects.”

[175] D. Birant and A. Kut, “ST-DBSCAN : An algorithm for clustering spatial – temporal data,” vol. 60, pp. 208–221, 2007, doi: 10.1016/j.datak.2006.01.013.

[176] S. J. Gaffney, A. W. Robertson, P. Smyth, S. J. Camargo, and M. Ghil, “Probabilistic Clustering of Extratropical Cyclones Using Regression Mixture Models.”

[177] J. Lee and J. Han, “Trajectory Clustering : A Partition-and-Group Framework ∗,” 2007.

[178] F. Schwenker and E. Trentin, “Pattern classification and clustering: A review of partially supervised learning approaches,” *Pattern Recognit. Lett.*, vol. 37, no. 1, pp. 4–14, 2014, doi: 10.1016/j.patrec.2013.10.017.

[179] S. B. Kotsiantis, I. D. Zaharakis, and P. E. Pintelas, “Machine learning: A review of classification and combining techniques,” *Artif. Intell. Rev.*, vol. 26, no. 3, pp. 159–190, 2006, doi: 10.1007/s10462-007-9052-3.

[180] A. Bolbol, T. Cheng, I. Tsapakis, and J. Haworth, “Inferring hybrid transportation modes from sparse GPS data using a moving window SVM classification,” *Comput. Environ. Urban Syst.*, vol. 36, no. 6, pp. 526–537, 2012, doi: 10.1016/j.compenvurbsys.2012.06.001.

[181] Y. Zheng, Y. Chen, Q. Li, X. Xie, and W. Y. Ma, “Understanding transportation modes based on GPS data for web applications,” *ACM Trans. Web*, vol. 4, no. 1, 2010, doi: 10.1145/1658373.1658374.

[182] D. Patel, C. Sheng, W. Hsu, and M. L. Lee, “Incorporating duration information for trajectory classification,” *Proc. - Int. Conf. Data Eng.*, pp. 1132–1143, 2012, doi: 10.1109/ICDE.2012.72.

[183] J. Y. Chemin, “Trajectory classification using hierarchical region-based and

trajectory-based clustering," *Commun. Pure Appl. Math.*, vol. 64, no. 12, pp. 1587–1598, 2011, doi: 10.1002/cpa.20386.

[184] J. D. Mazimpaka and S. Timpf, "Trajectory data mining: A review of methods and applications," *J. Spat. Inf. Sci.*, vol. 13, no. 2016, pp. 61–99, 2016, doi: 10.5311/josis.2016.13.263.

[185] S. Nasreen, M. A. Azam, K. Shehzad, U. Naeem, and M. A. Ghazanfar, "Frequent pattern mining algorithms for finding associated frequent patterns for data streams: A survey," *Procedia Comput. Sci.*, vol. 37, pp. 109–116, 2014, doi: 10.1016/j.procs.2014.08.019.

[186] M. Wachowicz, R. Ong, C. Renso, and M. Nanni, "Finding moving flock patterns among pedestrians through collective coherence," *Int. J. Geogr. Inf. Sci.*, vol. 25, no. 11, pp. 1849–1864, 2011, doi: 10.1080/13658816.2011.561209.

[187] H. Jeung, M. L. Yiu, X. Zhou, C. S. Jensen, and H. T. Shen, "Discovery of convoys in trajectory databases," *Proc. VLDB Endow.*, vol. 1, no. 1, pp. 1068–1080, 2008, doi: 10.14778/1453856.1453971.

[188] Z. Li, B. Ding, J. Han, and R. Kays, "Swarm: Mining relaxed temporal moving object clusters," *Proc. VLDB Endow.*, vol. 3, no. 1, pp. 723–734, 2010, doi: 10.14778/1920841.1920934.

[189] S. Sidiropoulos, K. Han, A. Majumdar, and W. Y. Ochieng, "Identifying significant traffic flow patterns in Multi-Airport systems terminal manoeuvring areas under uncertainty," *16th AIAA Aviat. Technol. Integr. Oper. Conf.*, no. June, pp. 1–19, 2016, doi: 10.2514/6.2016-3162.

[190] G. Antonini, J. P. Thiran, and S. Member, "Counting Pedestrians in Video Sequences Using Trajectory Clustering," vol. 16, no. 8, pp. 1008–1020, 2006.

[191] X. Olive and L. Basora, "Detection and identification of significant events in historical aircraft trajectory data," *Transp. Res. Part C*, vol. 119, no. August, p. 102737, 2020, doi: 10.1016/j.trc.2020.102737.

[192] X. Olive and L. Basora, “Deep Trajectory Clustering with Autoencoders,” 2020.

[193] A. Bombelli, L. Soler, E. Trumbauer, and K. D. Mease, “Strategic air traffic planning with fréchet distance aggregation and rerouting,” *J. Guid. Control. Dyn.*, vol. 40, no. 5, pp. 1117–1129, 2017, doi: 10.2514/1.G002308.

[194] H. Arneson, A. Bombelli, A. Segarra-Torne, and E. Tse, “Analysis of convective-weather impact on pre-departure routing decisions for flights traveling between Fort Worth Center and New York Air Center,” no. June, pp. 1–12, 2017, doi: 10.2514/6.2017-3593.

[195] P. Ren and L. Li, “Characterizing air traffic networks via large-scale aircraft tracking data: A comparison between China and the US networks,” *J. Air Transp. Manag.*, vol. 67, no. September 2017, pp. 181–196, 2018, doi: 10.1016/j.jairtraman.2017.12.005.

[196] Y. Liu, M. Hansen, M. O. Ball, D. J. Lovell, C. Chuang, and J. M. Gulding, “Causal analysis of en route flight inefficiency - The US experience,” *12th USA/Europe Air Traffic Manag. R D Semin.*, 2017.

[197] 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.*, vol. 42, no. 3, 2017, doi: 10.1145/3068335.

[198] A. M. Fahim, A. M. Salem, F. A. Torkey, and M. A. Ramadan, “Efficient enhanced k-means clustering algorithm,” *J. Zhejiang Univ. Sci.*, vol. 7, no. 10, pp. 1626–1633, 2006, doi: 10.1631/jzus.2006.A1626.

[199] 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, p. 7, 2020, doi: 10.3390/proceedings2020059007.

[200] A. Alharbi, A. Poujade, K. Malandrakis, I. Petrunin, D. Panagiotakopoulos, and A. Tsourdos, “Rule-based conflict management for unmanned traffic management scenarios,” *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, vol.

2020-Octob, 2020, doi: 10.1109/DASC50938.2020.9256690.

[201] A. P. Soares, “EXPLORING THE RANGE OF WEATHER IMPACTS ON UAS OPERATIONS,” *J. Chem. Inf. Model.*, vol. 53, no. 9, pp. 1689–1699, 2013.

[202] E. Capello, G. Guglieri, and F. Quagliotti, *A waypoint-based guidance algorithm for mini UAVs*, vol. 2, no. PART 1. IFAC, 2013.

[203] M. C. R. Murca and R. J. Hansman, “Identification, Characterization, and Prediction of Traffic Flow Patterns in Multi-Airport Systems,” *IEEE Trans. Intell. Transp. Syst.*, vol. 20, no. 5, pp. 1683–1696, 2019, doi: 10.1109/TITS.2018.2833452.

[204] M. Peng and W. Meng, “Cooperative Obstacle Avoidance for Multiple UAVs Using Spline_VO Method,” *Sensors*, vol. 22, no. 5, 2022, doi: 10.3390/s22051947.

[205] G. Gramajo and P. Shankar, “An Efficient Energy Constraint Based UAV Path Planning for Search and Coverage,” *Int. J. Aerosp. Eng.*, vol. 2017, 2017, doi: 10.1155/2017/8085623.

[206] M. Lalak and D. Wierzbicki, “Methodology of Detection and Classification of Selected Aviation Obstacles Based on UAV Dense Image Matching,” *IEEE J. Sel. Top. Appl. Earth Obs. Remote Sens.*, vol. 15, no. March, pp. 1869–1883, 2022, doi: 10.1109/JSTARS.2022.3149105.

[207] John K. Hall, “Algorithms and programs for the rapid computation of area and centre of mass,” *Comput. Geosci.*, vol. 1, no. 3, pp. 203–205, 1976.

[208] B. Hilburn, “Cognitive complexity in air traffic control : a literature review COGNITIVE COMPLEXITY IN AIR TRAFFIC CONTROL : A LITERATURE REVIEW Center for Human Performance Research,” no. November, 2017.

[209] A. Elgammal, R. Duraiswami, D. Harwood, and L. S. Davis, “Background and foreground modeling using nonparametric kernel density estimation for visual surveillance,” *Proc. IEEE*, vol. 90, no. 7, pp. 1151–1162, 2002, doi:

10.1109/JPROC.2002.801448.

[210] M. . J. Smith, G. Michael, A. P. Longley, and & Associates, *Geospatial Analysis, A Comprehensive Guide to Principles Techniques and Software Tools. 6th edition.* 2019.

[211] T. Ahola, K. Virrantaus, J. M. Krisp, and G. J. Hunter, “A spatio-temporal population model to support risk assessment and damage analysis for decision-making,” *Int. J. Geogr. Inf. Sci.*, vol. 21, no. 8, pp. 935–953, 2007, doi: 10.1080/13658810701349078.

[212] E. R. Mueller, P. H. Kopardekar, and K. H. Goodrich, “Enabling airspace integration for high-density on-demand mobility operations,” *17th AIAA Aviation Technology, Integration, and Operations Conference*. Denver, Colorado, p. 3086, 2017.

[213] International Civil Aviation Organization, *Doc. 4444 - Procedures for Air Navigation Services - Air Traffic Management*. 2016.

[214] D. Bertsimas and S. S. Patterson, “The Air Traffic Flow Management Problem with Enroute Capacities,” *Oper. Res.*, vol. 46, no. 3, pp. 406–422, 1998, doi: 10.1287/opre.46.3.406.

[215] A. M. F. Crespo, L. Weigang, and A. G. de Barros, “Reinforcement learning agents to tactical air traffic flow management,” *Int. J. Aviat. Manag.*, vol. 1, no. 3, pp. 145–161, 2012.

[216] T. Kistan, A. Gardi, R. Sabatini, S. Ramasamy, and E. Batuwangala, “An evolutionary outlook of air traffic flow management techniques,” *Prog. Aerosp. Sci.*, vol. 88, no. October 2016, pp. 15–42, 2017, doi: 10.1016/j.paerosci.2016.10.001.

[217] A. Gardi, R. Sabatini, and S. Ramasamy, “Multi-objective optimisation of aircraft flight trajectories in the ATM and avionics context,” *Prog. Aerosp. Sci.*, vol. 83, pp. 1–36, 2016, doi: 10.1016/j.paerosci.2015.11.006.

[218] Z. Zhao *et al.*, “Machine learning-based traffic management model for UAS

instantaneous density prediction in an urban area," *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, vol. 2020-Octob, 2020, doi: 10.1109/DASC50938.2020.9256471.

[219] Y. Lin, J. Zhang, and H. Liu, "Deep learning based short-term air traffic flow prediction considering temporal–spatial correlation," *Aerosp. Sci. Technol.*, vol. 93, p. 105113, 2019, doi: https://doi.org/10.1016/j.ast.2019.04.021.

[220] B. Hilburn, "Cognitive complexity in air traffic control: A literature review," Eurocontrol , Brussels, Belgium, 2004.

[221] E. M. Pfleiderer, C. A. Manning, and S. Goldman, "Relationship of complexity factor ratings with operational errors," no. May, 2007, [Online]. Available: http://oai.dtic.mil/oai/oai?verb=getRecord&metadataPrefix=html&identifier=ADA467731.

[222] D. Delahaye and S. Puechmorel, "Air traffic complexity based on dynamical systems," *Proc. IEEE Conf. Decis. Control*, pp. 2069–2074, 2010, doi: 10.1109/CDC.2010.5718004.

[223] J. M. Histon, R. J. Hansman, G. Aigoin, D. Delahaye, and S. Puechmorel, "Introducing Structural Considerations into Complexity Metrics," *Air Traffic Control Q.*, vol. 10, no. 2, 2002, doi: 10.2514/atcq.10.2.115.

[224] D. Delahaye and S. Puechmorel, *Modeling and Optimization of Air Traffic*. 2013.

[225] P. Juntama, S. Chaimatanan, S. Alam, and D. Delahaye, "A distributed metaheuristic approach for complexity reduction in air traffic for strategic 4d trajectory optimization," 2020, doi: 10.1109/AIDA-AT48540.2020.9049200.

[226] Z. Wang, D. Delahaye, J. L. Farges, and S. Alam, "Complexity optimal air traffic assignment in multi-layer transport network for Urban Air Mobility operations," *Transp. Res. Part C Emerg. Technol.*, vol. 142, no. April, 2022, doi: 10.1016/j.trc.2022.103776.

[227] D. Delahaye and S. Puechmorel, “Air traffic complexity: towards intrinsic metrics,” *3rd USA/Europe Air Traffic Manag. R&D Semin.*, no. June, 2000.

[228] Z. Wang, D. Delahaye, J. L. Farges, and S. Alam, “Air traffic assignment for intensive urban air mobility operations,” *J. Aerosp. Inf. Syst.*, vol. 18, no. 11, pp. 860–875, 2021, doi: 10.2514/1.I010954.

[229] Y. Lv, Y. Duan, W. Kang, Z. Li, and F. Y. Wang, “Traffic Flow Prediction With Big Data: A Deep Learning Approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 16, no. 2, pp. 865–873, 2015, doi: 10.1109/TITS.2014.2345663.

[230] M. Akhtar and S. Moridpour, “A Review of Traffic Congestion Prediction Using Artificial Intelligence,” *J. Adv. Transp.*, vol. 2021, pp. 1–18, 2021, doi: 10.1155/2021/8878011.

[231] H. Yuan and G. Li, “A Survey of Traffic Prediction: from Spatio-Temporal Data to Intelligent Transportation,” *Data Sci. Eng.*, vol. 6, no. 1, pp. 63–85, 2021, doi: 10.1007/s41019-020-00151-z.

[232] H. Liu, Y. Lin, Z. Chen, D. Guo, J. Zhang, and H. Jing, “Research on the Air Traffic Flow Prediction Using a Deep Learning Approach,” *IEEE Access*, vol. 7, pp. 148019–148030, 2019, doi: 10.1109/ACCESS.2019.2945821.

[233] Y. Lin, J. Zhang, and H. Liu, “An algorithm for trajectory prediction of flight plan based on relative motion between positions,” *Front. Inf. Technol. Electron. Eng.*, vol. 19, no. 7, pp. 905–916, 2018, doi: 10.1631/FITEE.1700224.

[234] W. Tian and M. Hu, “Study of air traffic flow management optimization model and algorithm based on multi-objective programming,” *ICCMS 2010 - 2010 Int. Conf. Comput. Model. Simul.*, vol. 2, pp. 210–214, 2010, doi: 10.1109/ICCMS.2010.20.

[235] S. Mehrmolaei and M. R. Keyvanpour, “Time series forecasting using improved ARIMA,” *2016 Artif. Intell. Robot. IRANOPEN 2016*, pp. 92–97, 2016, doi: 10.1109/RIOS.2016.7529496.

[236] E. Cadenas, W. Rivera, R. Campos-Amezcua, and C. Heard, “Wind speed prediction using a univariate ARIMA model and a multivariate NARX model,” *Energies*, vol. 9, no. 2, pp. 1–15, 2016, doi: 10.3390/en9020109.

[237] P. Xie, T. Li, J. Liu, S. Du, X. Yang, and J. Zhang, “Urban flow prediction from spatiotemporal data using machine learning: A survey,” *Inf. Fusion*, vol. 59, no. January, pp. 1–12, 2020, doi: 10.1016/j.inffus.2020.01.002.

[238] S. Sun, J. Chen, and J. Sun, “Traffic congestion prediction based on GPS trajectory data,” *Int. J. Distrib. Sens. Networks*, vol. 15, no. 5, 2019, doi: 10.1177/1550147719847440.

[239] M. G. Karlaftis and E. I. Vlahogianni, “Statistical methods versus neural networks in transportation research: Differences, similarities and some insights,” *Transp. Res. Part C Emerg. Technol.*, vol. 19, no. 3, pp. 387–399, 2011, doi: https://doi.org/10.1016/j.trc.2010.10.004.

[240] K. M. Nadeem and T. P. Fowdur, “Performance analysis of a real-time adaptive prediction algorithm for traffic congestion,” *J. Inf. Commun. Technol.*, vol. 17, no. 3, pp. 493–511, 2018, doi: 10.32890/jict2018.17.3.8262.

[241] T. Ito and R. Kaneyasu, “Predicting traffic congestion using driver behavior,” *Procedia Comput. Sci.*, vol. 112, pp. 1288–1297, 2017, doi: 10.1016/j.procs.2017.08.090.

[242] H. H. Zhang, C. P. Jiang, and L. Yang, “Forecasting traffic congestion status in terminal areas based on support vector machine,” *Adv. Mech. Eng.*, vol. 8, no. 9, pp. 1–11, 2016, doi: 10.1177/1687814016667384.

[243] S. Reza, H. S. Oliveira, J. J. M. Machado, and J. M. R. S. Tavares, *Urban safety: An image-processing and deep-learning-based intelligent traffic management and control system*, vol. 21, no. 22. 2021.

[244] Z. Zhao, W. Chen, X. Wu, P. C. Y. Chen, and J. Liu, “LSTM network: A deep learning approach for Short-term traffic forecast,” *IET Intell. Transp. Syst.*, vol. 11, no. 2, pp. 68–75, 2017, doi: 10.1049/iet-its.2016.0208.

[245] Y. LeCun, L. Bottou, Y. Bengio, and P. Haffner, “Gradient-based learning applied to document recognition,” *Proc. IEEE*, vol. 86, no. 11, pp. 2278–2323, 1998, doi: 10.1109/5.726791.

[246] I. Sutskever, O. Vinyals, and Q. V Le, “Sequence to sequence learning with neural networks,” *Adv. Neural Inf. Process. Syst.*, vol. 4, no. 1, pp. 3104–3112, 2014.

[247] A. Géron, *Hands-on machine learning with Scikit-Learn, Keras, and TensorFlow*. O’Reilly Media, Inc., 2022.

[248] J. Rosindell and Y. Wong, “On the difficulty of training recurrent neural networks,” *Phylogenetic Divers. Appl. Challenges Biodivers. Sci.*, no. 2, pp. 41–71, 2018, doi: 10.1007/978-3-319-93145-6_3.

[249] S. Hochreiter and J. Schmidhuber, “Long Short-Term Memory,” *Neural Comput.*, vol. 9, no. 8, pp. 1735–1780, 1997, doi: 10.1162/neco.1997.9.8.1735.

[250] R. C. Staudemeyer and E. R. Morris, “Understanding LSTM--a tutorial into long short-term memory recurrent neural networks,” *arXiv Prepr. arXiv1909.09586*, pp. 1–42, 2019.

[251] H. Yu, Z. Wu, S. Wang, Y. Wang, and X. Ma, “Spatiotemporal recurrent convolutional networks for traffic prediction in transportation networks,” *Sensors (Switzerland)*, vol. 17, no. 7, pp. 1–16, 2017, doi: 10.3390/s17071501.

[252] W. Zhang, Y. Yu, Y. Qi, F. Shu, and Y. Wang, “Short-term traffic flow prediction based on spatio-temporal analysis and CNN deep learning,” *Transp. A Transp. Sci.*, vol. 15, no. 2, pp. 1688–1711, 2019, doi: 10.1080/23249935.2019.1637966.

[253] T. Bogaerts, A. D. Masegosa, J. S. Angarita-Zapata, E. Onieva, and P. Hellinckx, “A graph CNN-LSTM neural network for short and long-term traffic forecasting based on trajectory data,” *Transp. Res. Part C Emerg. Technol.*, vol. 112, no. October, pp. 62–77, 2020, doi:

10.1016/j.trc.2020.01.010.

[254] Z. Wu, S. Pan, G. Long, J. Jiang, X. Chang, and C. Zhang, “Connecting the Dots: Multivariate Time Series Forecasting with Graph Neural Networks,” *Proc. ACM SIGKDD Int. Conf. Knowl. Discov. Data Min.*, pp. 753–763, 2020, doi: 10.1145/3394486.3403118.

[255] W. Jiang and J. Luo, “Graph neural network for traffic forecasting: A survey,” *Expert Syst. Appl.*, vol. 207, no. May, pp. 1–28, 2022, doi: 10.1016/j.eswa.2022.117921.

[256] K. Cai, Z. Shen, X. Luo, and Y. Li, “Temporal attention aware dual-graph convolution network for air traffic flow prediction,” *J. Air Transp. Manag.*, vol. 106, no. August 2022, p. 102301, 2023, doi: 10.1016/j.jairtraman.2022.102301.

[257] Y. Jiang *et al.*, “Spatial-Temporal Graph Data Mining for IoT-Enabled Air Mobility Prediction,” *IEEE Internet Things J.*, vol. 9, no. 12, pp. 9232–9240, 2022, doi: 10.1109/JIOT.2021.3090265.

[258] L. Shi-Garrier, D. Delahaye, and N. C. Bouaynaya, “Predicting Air Traffic Congested Areas with Long Short-Term Memory Networks.”

[259] W. Wei, H. Wu, and H. Ma, “An AutoEncoder and LSTM-Based Traffic Flow Prediction Method,” *Sensors*, vol. 19, no. 13. 2019, doi: 10.3390/s19132946.

[260] R. Fu, Z. Zhang, and L. Li, “Using LSTM and GRU neural network methods for traffic flow prediction,” in *2016 31st Youth Academic Annual Conference of Chinese Association of Automation (YAC)*, 2016, pp. 324–328, doi: 10.1109/YAC.2016.7804912.

[261] D. Delahaye, A. García, J. Lavandier, S. Chaimatanan, and M. Soler, “Air traffic complexity map based on linear dynamical systems,” *Aerospace*, vol. 9, no. 5, pp. 1–17, 2022, doi: 10.3390/aerospace9050230.

[262] M. Ribeiro, J. Ellerbroek, and J. Hoekstra, “Analysis of conflict resolution

methods for manned and unmanned aviation using fast-time simulations," *SESAR Innov. Days*, no. December, 2019.

[263] H. Widiputra, A. Mailangkay, and E. Gautama, "Multivariate CNN-LSTM Model for Multiple Parallel Financial Time-Series Prediction," *Complexity*, vol. 2021, 2021, doi: 10.1155/2021/9903518.

[264] L. Lu, X. Zhang, and S. Renais, "On training the recurrent neural network encoder-decoder for large vocabulary end-to-end speech recognition," in *2016 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, 2016, pp. 5060–5064, doi: 10.1109/ICASSP.2016.7472641.

[265] S. Rafi and R. Das, "RNN Encoder and Decoder with Teacher Forcing Attention Mechanism for Abstractive Summarization," *Proc. 2021 IEEE 18th India Counc. Int. Conf. INDICON 2021*, 2021, doi: 10.1109/INDICON52576.2021.9691681.

[266] R. J. Williams and D. Zipser, "A Learning Algorithm for Continually Running Fully Recurrent Neural Networks," *Neural Comput.*, vol. 1, no. 2, pp. 270–280, 1989, doi: 10.1162/neco.1989.1.2.270.

[267] D. P. Kingma and J. L. Ba, "Adam: A method for stochastic optimization," *3rd Int. Conf. Learn. Represent. ICLR 2015 - Conf. Track Proc.*, pp. 1–15, 2015.

[268] A. Alharbi, I. Petrunin, and D. Panagiotakopoulos, "Traffic Flow Prediction for UTM Application: A Deep Learning Approach," *2022 IEEE/AIAA 41st Digit. Avion. Syst. Conf.*, pp. 1–10, Sep. 2022, doi: 10.1109/DASC55683.2022.9925774.

[269] R. Geng, D. Cui, and B. Xu, "Support vector machine-based combinational model for air traffic forecasts," *J. Tsinghua Univ. (Science Technol.*, vol. 48, no. 7, pp. 1205–1208, 2008.

[270] Q. F. Li and Z. M. Song, "High-performance concrete strength prediction based on ensemble learning," *Constr. Build. Mater.*, vol. 324, no. February,

p. 126694, 2022, doi: 10.1016/j.conbuildmat.2022.126694.

[271] Z. Zhao *et al.*, “Machine learning-based traffic management model for UAS instantaneous density prediction in an urban area,” *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, vol. 2020-Octob, 2020, doi: 10.1109/DASC50938.2020.9256471.

[272] G. Zhong, H. Zhang, J. Zhou, J. Zhou, and H. Liu, “Short-Term 4D Trajectory Prediction for UAV Based on Spatio-Temporal Trajectory Clustering,” *IEEE Access*, vol. 10, pp. 93362–93380, 2022, doi: 10.1109/ACCESS.2022.3203428.

[273] S. F. Abedin, M. S. Munir, N. H. Tran, Z. Han, and C. S. Hong, “Data Freshness and Energy-Efficient UAV Navigation Optimization: A Deep Reinforcement Learning Approach,” *IEEE Trans. Intell. Transp. Syst.*, vol. 22, no. 9, pp. 5994–6006, 2021, doi: 10.1109/TITS.2020.3039617.

[274] R. Komatsu, A. Aurilla, A. Bechina, S. Güldal, and M. Şaşmaz, “Machine Learning Attempt to Conflict Detection for UAV with System Failure in U-Space: Recurrent Neural Network, RNNn; Machine Learning Attempt to Conflict Detection for UAV with System Failure in U-Space: Recurrent Neural Network, RNNn,” 2022, doi: 10.1109/ICUAS54217.2022.9836147.

[275] S. Yang, Z. Meng, X. Chen, and R. Xie, “Real-time obstacle avoidance with deep reinforcement learning three-dimensional autonomous obstacle avoidance for uav,” in *RICAI 2019: International Conference on Robotics, Intelligent Control and Artificial Intelligence*, pp. 324–329.

[276] Z. Zhang *et al.*, “Neural Network Architecture Search and Model Compression for Fast Prediction of UAS Traffic Density,” pp. 1–9, 2021.

[277] S. Kiranyaz, O. Avci, O. Abdeljaber, T. Ince, M. Gabbouj, and D. J. Inman, “1D convolutional neural networks and applications: A survey,” *Mech. Syst. Signal Process.*, vol. 151, p. 107398, 2021, doi: 10.1016/j.ymssp.2020.107398.

[278] A. Barredo Arrieta *et al.*, “Explainable Artificial Intelligence (XAI): Concepts,

taxonomies, opportunities and challenges toward responsible AI," *Inf. Fusion*, vol. 58, pp. 82–115, 2020, doi: https://doi.org/10.1016/j.inffus.2019.12.012.

[279] Y. Xie, N. Pongsakornsathien, A. Gardi, and R. Sabatini, "Explanation of machine-learning solutions in air-traffic management," *Aerospace*, vol. 8, no. 8, 2021, doi: 10.3390/aerospace8080224.

[280] Y. Kim, J. Jo, and M. Shaw, "A lightweight communication architecture for small UAS Traffic Management (sUTM)," *ICNS 2015 - Innov. Oper. Implement. Benefits Integr. CNS Infrastructure, Conf. Proc.*, pp. 1–9, 2015, doi: 10.1109/ICNSURV.2015.7121342.

[281] Y. Tang, Y. Xu, and G. Inalhan, "An Integrated Approach for On-Demand Dynamic Capacity Management Service in U-Space," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 58, no. 5, pp. 4180–4195, 2022, doi: 10.1109/TAES.2022.3159317.

[282] C. Capitán, H. Pérez-León, J. Capitán, Á. Castaño, and A. Ollero, "Unmanned Aerial Traffic Management System Architecture for U-Space In-Flight Services," *Appl. Sci.*, vol. 11, no. 9, p. 3995, 2021, doi: 10.3390/app11093995.

[283] P. S. Escalonilla, D. Janisch, C. Forster, M. Büddefeld, and H. E. Teomitzi, "Towards a continuous Demand and Capacity. Balancing process for U-space."

[284] C. Chin, K. Gopalakrishnan, H. Balakrishnan, M. Egorov, and A. Evans, "Tradeoffs between Efficiency and Fairness in Unmanned Aircraft Systems Traffic Management," *9th Int. Conf. Res. Air Transp.*, pp. 1–8, 2020.

[285] D. Bertsimas and S. S. Patterson, "The air traffic flow management problem with enroute capacities," *Oper. Res.*, vol. 46, no. 3, pp. 406–422, 1998, doi: 10.1287/opre.46.3.406.

[286] A. M. F. Crespo, L. Weigang, and A. G. de Barros, "Reinforcement learning agents to tactical air traffic flow management," *Int. J. Aviat. Manag.*, vol. 1,

no. 3, p. 145, 2012, doi: 10.1504/ijam.2012.045736.

[287] T. Kistan, A. Gardi, R. Sabatini, S. Ramasamy, and E. Batuwangala, “An evolutionary outlook of air traffic flow management techniques,” *Prog. Aerosp. Sci.*, vol. 88, no. October 2016, pp. 15–42, 2017, doi: 10.1016/j.paerosci.2016.10.001.

[288] A. Gardi, R. Sabatini, and S. Ramasamy, “Multi-objective optimisation of aircraft flight trajectories in the ATM and avionics context,” *Prog. Aerosp. Sci.*, vol. 83, pp. 1–36, 2016, doi: 10.1016/j.paerosci.2015.11.006.

[289] A. Degas *et al.*, “A Survey on Artificial Intelligence (AI) and eXplainable AI in Air Traffic Management: Current Trends and Development with Future Research Trajectory,” *Appl. Sci.*, vol. 12, no. 3, pp. 1–47, 2022, doi: 10.3390/app12031295.

[290] T. Kistan, A. Gardi, and R. Sabatini, “Machine learning and cognitive ergonomics in air traffic management: Recent developments and considerations for certification,” *Aerospace*, vol. 5, no. 4, 2018, doi: 10.3390/aerospace5040103.

[291] C. Borst, V. A. Bijsterbosch, M. M. van Paassen, and M. Mulder, “Ecological interface design: supporting fault diagnosis of automated advice in a supervisory air traffic control task,” *Cogn. Technol. Work*, vol. 19, no. 4, pp. 545–560, 2017, doi: 10.1007/s10111-017-0438-y.

[292] G. Borghini *et al.*, “EEG-Based Cognitive Control Behaviour Assessment: An Ecological study with Professional Air Traffic Controllers,” *Sci. Rep.*, vol. 7, no. 1, pp. 1–16, 2017, doi: 10.1038/s41598-017-00633-7.

[293] R. Sheh and I. Monteath, “Defining Explainable AI for Requirements Analysis,” *KI - Kunstl. Intelligenz*, vol. 32, no. 4, pp. 261–266, 2018, doi: 10.1007/s13218-018-0559-3.

[294] S. T. Mueller, R. R. Hoffman, W. Clancey, A. Emrey, and G. Klein, “Explanation in Human-AI Systems: A Literature Meta-Review,” *Def. Adv. Res. Proj. Agency*, no. February 2019, p. 204, 2019.

[295] J. R. Quinlan, “Simplifying decision trees,” *Int. J. Hum. Comput. Stud.*, vol. 51, no. 2, pp. 497–510, 1999, doi: 10.1006/ijhc.1987.0321.

[296] C. Gokceoglu, H. A. Nefeslioglu, E. Sezer, A. S. Bozkir, and T. Y. Duman, “Assessment of landslide susceptibility by decision trees in the metropolitan area of Istanbul, Turkey,” *Math. Probl. Eng.*, vol. 2010, 2010, doi: 10.1155/2010/901095.

[297] D. Choi, S., Kim, Y. J., Briceno, S., & Mavris, “Prediction of weather-induced airline delays based on machine learning algorithms,” *AIAA/IEEE Digit. Avion. Syst. Conf.*, pp. 1–6, 2016.

[298] L. Carvalho *et al.*, “On the relevance of data science for flight delay research: a systematic review,” *Transp. Rev.*, vol. 0, no. 0, pp. 1–30, 2020, doi: 10.1080/01441647.2020.1861123.

[299] Y. Tian, B. Ye, L. Wan, M. Yang, and D. Xing, “Restricted airspace unit identification using density-based spatial clustering of applications with noise,” *Sustain.*, vol. 11, no. 21, 2019, doi: 10.3390/su11215962.

[300] S. Pouyanfar *et al.*, “A Survey on Deep Learning,” *ACM Comput. Surv.*, vol. 51, no. 5, pp. 1–36, 2019, doi: 10.1145/3234150.

[301] A. Sternberg, D. Carvalho, L. Murta, J. Soares, and E. Ogasawara, “An analysis of Brazilian flight delays based on frequent patterns,” *Transp. Res. Part E Logist. Transp. Rev.*, vol. 95, pp. 282–298, 2016, doi: 10.1016/j.tre.2016.09.013.

[302] J. Cheng, C. Rong, H. Ye, and X. Zheng, “Risk management using big real time data,” *Proc. - IEEE 7th Int. Conf. Cloud Comput. Technol. Sci. CloudCom 2015*, pp. 542–547, 2016, doi: 10.1109/CloudCom.2015.103.

[303] R. Arnaldo Scarpel and L. C. Pelicioni, “A data analytics approach for anticipating congested days at the São Paulo International Airport,” *J. Air Transp. Manag.*, vol. 72, no. July, pp. 1–10, 2018, doi: 10.1016/j.jairtraman.2018.07.002.

[304] J. Cheng, “Estimation of flight delay using weighted Spline combined with ARIMA model,” *Proc. 2014 IEEE 7th Int. Conf. Adv. Infocomm Technol. IEEE/ICAIT 2014*, pp. 8–20, 2015, doi: 10.1109/ICAIT.2014.7019523.

[305] J. Oehling and D. J. Barry, “Using machine learning methods in airline flight data monitoring to generate new operational safety knowledge from existing data,” *Saf. Sci.*, vol. 114, no. January, pp. 89–104, 2019, doi: 10.1016/j.ssci.2018.12.018.

[306] C. Büsing, D. Kadatz, and C. Cleophas, “Capacity uncertainty in airline revenue management: Models, algorithms, and computations,” *Transp. Sci.*, vol. 53, no. 2, pp. 383–400, 2019, doi: 10.1287/trsc.2018.0829.

[307] S. H. Chung, H. L. Ma, and H. K. Chan, “Cascading Delay Risk of Airline Workforce Deployments with Crew Pairing and Schedule Optimization,” *Risk Anal.*, vol. 37, no. 8, pp. 1443–1458, 2017, doi: 10.1111/risa.12746.

[308] Y. Liu, Y. Liu, M. Hansen, A. Pozdnukhov, and D. Zhang, “Using machine learning to analyze air traffic management actions: Ground delay program case study,” *Transp. Res. Part E Logist. Transp. Rev.*, vol. 131, no. December 2018, pp. 80–95, 2019, doi: 10.1016/j.tre.2019.09.012.

[309] Y. Yanying, H. Mo, and L. Haifeng, “Classification Prediction Analysis of Flight Cancellation Based on Spark,” *Procedia Comput. Sci.*, vol. 162, no. Itqm 2019, pp. 480–486, 2020, doi: 10.1016/j.procs.2019.12.014.

[310] C. Torens, F. Jünger, S. Schirmer, S. Schopferer, T. Maienschein, and J. C. Dauer, “Machine Learning Verification and Safety for Unmanned Aircraft – A Literature Study,” *AIAA Sci. Technol. Forum Expo. AIAA SciTech Forum 2022*, 2022, doi: 10.2514/6.2022-1133.

[311] C. Krittanawong, H. J. Zhang, Z. Wang, M. Aydar, and T. Kitai, “Artificial Intelligence in Precision Cardiovascular Medicine,” *J. Am. Coll. Cardiol.*, vol. 69, no. 21, pp. 2657–2664, 2017, doi: 10.1016/j.jacc.2017.03.571.

[312] B. M. Keneni *et al.*, “Evolving Rule-Based Explainable Artificial Intelligence for Unmanned Aerial Vehicles,” *IEEE Access*, vol. 7, pp. 17001–17016,

2019, doi: 10.1109/ACCESS.2019.2893141.

[313] Y. Xie, N. Pongsakornsathien, A. Gardi, and R. Sabatini, “Explanation of Machine-Learning Solutions in Air-Traffic Management,” *Aerospace*, vol. 8, no. 8. 2021, doi: 10.3390/aerospace8080224.

[314] J. M. Faria, “Machine learning safety: An overview,” *Proc. 26th Safety-Critical Syst. Symp. York, UK*, no. October 2017, pp. 6–8, 2018.

[315] F. Tambon *et al.*, “How to certify machine learning based safety-critical systems? A systematic literature review,” *Autom. Softw. Eng.*, vol. 29, no. 2, pp. 1–92, 2022, doi: 10.1007/s10515-022-00337-x.

[316] D. Amodei, C. Olah, J. Steinhardt, P. Christiano, J. Schulman, and D. Mané, “Concrete Problems in AI Safety,” pp. 1–29, 2016, [Online]. Available: http://arxiv.org/abs/1606.06565.

[317] A. C. Serban, “Designing Safety Critical Software Systems to Manage Inherent Uncertainty,” *Proc. - 2019 IEEE Int. Conf. Softw. Archit. - Companion, ICSA-C 2019*, pp. 246–249, 2019, doi: 10.1109/ICSA-C.2019.00051.

[318] G. Hains, A. Jakobsson, and Y. Khmelevsky, “Towards formal methods and software engineering for deep learning: Security, safety and productivity for dl systems development,” *12th Annu. IEEE Int. Syst. Conf. SysCon 2018 - Proc.*, pp. 1–5, 2018, doi: 10.1109/SYSCON.2018.8369576.

[319] Bharadwaj, K. B. Prakash, and G. R. Kanagachidambaresan, *Pattern Recognition and Machine Learning*. 2021.

[320] P. Rodríguez-Dapena, “Software safety certification: A multidomain problem,” *IEEE Softw.*, vol. 16, no. 4, pp. 31–38, 1999, doi: 10.1109/52.776946.

[321] W. K. Youn, S. B. Hong, K. R. Oh, and O. S. Ahn, “Software certification of safety-critical avionic systems: DO-178C and its impacts,” *IEEE Aerosp. Electron. Syst. Mag.*, vol. 30, no. 4, pp. 4–13, 2015, doi:

10.1109/MAES.2014.140109.

[322] A. Pereira and C. Thomas, “Challenges of Machine Learning Applied to Safety-Critical Cyber-Physical Systems,” *Mach. Learn. Knowl. Extr.*, vol. 2, no. 4, pp. 579–602, 2020, doi: 10.3390/make2040031.

[323] W. Liu, Z. Wang, X. Liu, N. Zeng, Y. Liu, and F. E. Alsaadi, “A Survey of Deep Neural Network Architectures and Their Applications,” *Elsevier*, pp. 1–31, 2017, [Online]. Available: https://www.sciencedirect.com/science/article/pii/S0925231216315533.

[324] S. Pouyanfar *et al.*, “A survey on deep learning: Algorithms, techniques, and applications,” *ACM Comput. Surv.*, vol. 51, no. 5, 2018, doi: 10.1145/3234150.

[325] S. Padakandla, “A Survey of Reinforcement Learning Algorithms for Dynamically Varying Environments,” *ACM Comput. Surv.*, vol. 54, no. 6, 2021, doi: 10.1145/3459991.

[326] K. Arulkumaran, M. P. Deisenroth, M. Brundage, and A. A. Bharath, “Deep reinforcement learning: A brief survey,” *IEEE Signal Process. Mag.*, vol. 34, no. 6, pp. 26–38, 2017, doi: 10.1109/MSP.2017.2743240.

[327] J. Lu, V. Behbood, P. Hao, H. Zuo, S. Xue, and G. Zhang, “Transfer learning using computational intelligence: A survey,” *Knowledge-Based Syst.*, vol. 80, pp. 14–23, 2015, doi: 10.1016/j.knosys.2015.01.010.

[328] K. Weiss, T. M. Khoshgoftaar, and D. D. Wang, *A survey of transfer learning*, vol. 3, no. 1. Springer International Publishing, 2016.

[329] H. M. Gomes, J. P. Barddal, A. F. Enembreck, and A. Bifet, “A survey on ensemble learning for data stream classification,” *ACM Comput. Surv.*, vol. 50, no. 2, 2017, doi: 10.1145/3054925.

[330] B. Krawczyk, L. L. Minku, J. Gama, J. Stefanowski, and M. Woźniak, “Ensemble learning for data stream analysis: A survey,” *Inf. Fusion*, vol. 37, pp. 132–156, 2017, doi: 10.1016/j.inffus.2017.02.004.

[331] R. Ashmore, R. Calinescu, and C. Paterson, “Assuring the Machine Learning Lifecycle: Desiderata, Methods, and Challenges,” *ACM Comput. Surv.*, vol. 54, no. 5, 2021, doi: 10.1145/3453444.

[332] R. Hawkins, C. Paterson, C. Picardi, Y. Jia, R. Calinescu, and I. Habli, “Guidance on the Assurance of Machine Learning in Autonomous Systems (AMLAS),” no. 1, 2021, [Online]. Available: http://arxiv.org/abs/2102.01564.

[333] C. S. Hernandez, S. Ayo, and D. Panagiotakopoulos, “An Explainable Artificial Intelligence (xAI) Framework for Improving Trust in Automated ATM Tools,” *AIAA/IEEE Digit. Avion. Syst. Conf. - Proc.*, vol. 2021-Octob, pp. 3–7, 2021, doi: 10.1109/DASC52595.2021.9594341.

[334] E. Artificial, I. Roadmap, E. Artificial, and I. Roadmap, “A human-centric approach to AI in aviation,” no. February, 2020.

[335] C. Molnar, “Interpretable Machine Learning A Guide for Making Black Box Models Explainable.”

[336] G. Marcus, “Deep Learning : A Critical Appraisal,” pp. 1–27.

[337] A. Preece, D. Harborne, D. Braines, R. Tomsett, and S. Chakraborty, “Stakeholders in Explainable AI,” 2018, [Online]. Available: http://arxiv.org/abs/1810.00184.

[338] D.Gunning, “DARPA’s Explainable Artificial Intelligence Program,” *Def. Adv. Res. Proj. Agency (DARPA*, pp. 341–368, 2017, doi: 10.1201/b10933-22.

[339] E. Tjoa and C. Guan, “A Survey on Explainable Artificial Intelligence (XAI): Toward Medical XAI,” *IEEE Trans. Neural Networks Learn. Syst.*, vol. 32, no. 11, pp. 4793–4813, 2021, doi: 10.1109/TNNLS.2020.3027314.

[340] J. Zhu, A. Liapis, S. Risi, R. Bidarra, and G. M. Youngblood, “Explainable AI for Designers: A Human-Centered Perspective on Mixed-Initiative Co-Creation,” *IEEE Conf. Comput. Intell. Games, CIG*, vol. 2018-Augus, pp.

1–8, 2018, doi: 10.1109/CIG.2018.8490433.

[341] F. K. Dosilovic, M. Brcic, and N. Hlupic, “Explainable artificial intelligence: A survey,” in *2018 41st International Convention on Information and Communication Technology, Electronics and Microelectronics, MIPRO 2018 - Proceedings*, 2018, pp. 210–215, doi: 10.23919/MIPRO.2018.8400040.

[342] A. Barredo Arrieta *et al.*, “Explainable Artificial Intelligence (XAI): Concepts, taxonomies, opportunities and challenges toward responsible AI,” *Inf. Fusion*, vol. 58, no. October 2019, pp. 82–115, 2020, doi: 10.1016/j.inffus.2019.12.012.

[343] Z. C. Lipton, “The Mythos of Model Interpretability: In Machine Learning, the Concept of Interpretability Is Both Important and Slippery,” vol. 16, no. 3, pp. 31–57, 2018, doi: 10.1145/3236386.3241340.

[344] J.-P. Kucklick, “Towards a model- and data-focused taxonomy of XAI systems,” *Wirtschaftsinformatik 2022 Proc.*, 2022, [Online]. Available: https://aisel.aisnet.org/wi2022/business_analytics/business_analytics/2.

[345] M. Flora, C. Potvin, A. McGovern, and S. Handler, “Comparing Explanation Methods for Traditional Machine Learning Models Part 1: An Overview of Current Methods and Quantifying Their Disagreement,” pp. 1–22, 2022, [Online]. Available: http://arxiv.org/abs/2211.08943.

[346] L. Hyafil and R. L. Rivest, “Constructing Optimal Binary Trees in NP-Complete,” *Information Processing Letters*. pp. 1–3, 1976, [Online]. Available: https://people.csail.mit.edu/rivest/HyafilRivest-ConstructingOptimalBinaryDecisionTreesIsNPComplete.pdf.

[347] A. C. H. Núñez, C. Angulo, “Rule extraction from support vector machines,” p. 520, 2002.

[348] U. Johansson, R. König, and L. Niklasson, “The truth is in there - Rule extraction from opaque models using genetic programming,” *Proc. Seventeenth Int. Florida Artif. Intell. Res. Soc. Conf. FLAIRS 2004*, vol. 2,

no. February 2014, pp. 658–663, 2004.

[349] C. Rudin, “Stop explaining black box machine learning models for high stakes decisions and use interpretable models instead,” *Nat. Mach. Intell.*, vol. 1, no. 5, pp. 206–215, 2019, doi: 10.1038/s42256-019-0048-x.

[350] M. Lechner, R. Hasani, A. Amini, T. A. Henzinger, D. Rus, and R. Grosu, “Neural circuit policies enabling auditable autonomy,” *Nat. Mach. Intell.*, vol. 2, no. 10, pp. 642–652, 2020, doi: 10.1038/s42256-020-00237-3.

[351] M. T. Ribeiro, S. Singh, and C. Guestrin, “‘Why Should I Trust You?’ Explaining the Predictions of Any Classifier,” *NAACL-HLT 2016 - 2016 Conf. North Am. Chapter Assoc. Comput. Linguist. Hum. Lang. Technol. Proc. Demonstr. Sess.*, pp. 97–101, 2016, doi: 10.18653/v1/n16-3020.

[352] S.-I. Lundberg, S. M., and Lee, “A Unified Approach to Interpreting Model Predictions,” *Proc. 31st Int. Conf. Neural Inf. Process. Syst. Curran Assoc. Inc., Long Beach, California, USA, 2017, pp. 4768–4777. doi10.5555/3295222.3295230*, pp. 4768–4777, Apr. 2017, doi: 10.5555/3295222.3295230.

[353] E. Strumbelj and I. Kononenko, “An Efficient Explanation of Individual Classifications using Game Theory,” vol. 11, pp. 1–18, 2010.

[354] S. M. Lundberg *et al.*, “From local explanations to global understanding with explainable AI for trees,” *Nat. Mach. Intell.*, vol. 2, no. January, 2020, doi: 10.1038/s42256-019-0138-9.

[355] M. D. Zeiler and R. Fergus, “Visualizing and understanding convolutional networks,” *Lect. Notes Comput. Sci. (including Subser. Lect. Notes Artif. Intell. Lect. Notes Bioinformatics)*, vol. 8689 LNCS, no. PART 1, pp. 818–833, 2014, doi: 10.1007/978-3-319-10590-1_53.

[356] K. Simonyan, A. Vedaldi, and A. Zisserman, “Deep inside convolutional networks: Visualising image classification models and saliency maps,” *2nd Int. Conf. Learn. Represent. ICLR 2014 - Work. Track Proc.*, pp. 1–8, 2014.

[357] R. R. Selvaraju, M. Cogswell, A. Das, R. Vedantam, D. Parikh, and D. Batra, “Grad-CAM: Visual Explanations from Deep Networks via Gradient-Based Localization,” *Int. J. Comput. Vis.*, vol. 128, no. 2, pp. 336–359, 2020, doi: 10.1007/s11263-019-01228-7.

[358] T. Lei, R. Barzilay, and T. Jaakkola, “Rationalizing neural predictions,” *EMNLP 2016 - Conf. Empir. Methods Nat. Lang. Process. Proc.*, pp. 107–117, 2016, doi: 10.18653/v1/d16-1011.

[359] B. Kim *et al.*, “Interpretability beyond feature attribution: Quantitative Testing with Concept Activation Vectors (TCAV),” *35th Int. Conf. Mach. Learn. ICML 2018*, vol. 6, pp. 4186–4195, 2018.

[360] J. B. Lamy, B. Sekar, G. Guezennec, J. Bouaud, and B. Séroussi, “Explainable artificial intelligence for breast cancer: A visual case-based reasoning approach,” *Artif. Intell. Med.*, vol. 94, no. December 2018, pp. 42–53, 2019, doi: 10.1016/j.artmed.2019.01.001.

[361] R. R. Hoffman, S. T. Mueller, G. Klein, and J. Litman, “Metrics for Explainable AI: Challenges and Prospects,” pp. 1–50, 2018, [Online]. Available: http://arxiv.org/abs/1812.04608.

[362] S. Mohseni, N. Zarei, and E. D. Ragan, “A Multidisciplinary Survey and Framework for Design and Evaluation of Explainable AI Systems,” *ACM Trans. Interact. Intell. Syst.*, vol. 11, no. 3–4, pp. 1–45, 2021, doi: 10.1145/3387166.

[363] F. Doshi-Velez and B. Kim, “Towards A Rigorous Science of Interpretable Machine Learning,” no. Ml, pp. 1–13, 2017, [Online]. Available: http://arxiv.org/abs/1702.08608.

[364] C. Barnhart, D. Fearing, A. Odoni, and V. Vaze, “Demand and capacity management in air transportation,” *EURO J. Transp. Logist.*, vol. 1, no. 1–2, pp. 135–155, 2012, doi: 10.1007/s13676-012-0006-9.

[365] G. Flynn, A. Benkouar, and R. Christien, “Pessimistic Sector Capacity Estimation,” *EEC Note*, vol. 21, no. 03, p. 1, 2003, [Online]. Available:

http://scholar.google.com/scholar?hl=en&btnG=Search&q=intitle:REvisiting+the+Swiss+Cheese+model+of+accidents#0.

[366] J. C. Allen, “Sample size calculation for two independent groups: A useful rule of thumb,” *Proc. Singapore Healthc.*, vol. 20, no. 2, pp. 138–140, 2011, doi: 10.1177/201010581102000213.

[367] A. Alharbi, I. Petrunin, and D. Panagiotakopoulos, “Modeling and Characterization of Traffic Flow Patterns and Identification of Airspace Density for UTM application,” *IEEE Access*, pp. 1–26, 2022, doi: 10.1109/ACCESS.2022.3228828.

[368] I. Grau, D. Sengupta, M. Matilde, G. Lorenzo, and A. Nowe, “Grey-Box Model: An ensemble approach for addressing semi-supervised classification problems,” *Benelearn 2016 Belgian-Dutch Conf. Mach. Learn.*, 2016, [Online]. Available: http://archive.ics.uci.edu/ml.

[369] M. Robnik-Šikonja and I. Kononenko, “Explaining classifications for individual instances,” *IEEE Trans. Knowl. Data Eng.*, vol. 20, no. 5, pp. 589–600, 2008, doi: 10.1109/TKDE.2007.190734.

[370] E. Pintelas, I. E. Livieris, and P. Pintelas, “A Grey-Box ensemble model exploiting Black-Box accuracy and White-Box intrinsic interpretability,” *Algorithms*, vol. 13, no. 1, pp. 1–17, 2020, doi: 10.3390/a13010017.

[371] M. Abdellatief, M. Sultan, and A. Bakar, “Component-based Software System Dependency Metrics based on Component Information Flow Measurements,” *… Softw. …*, no. c, pp. 76–83, 2011, [Online]. Available: http://www.thinkmind.org/index.php?view=article&articleid=icsea_2011_4_10_10260.

[372] S. Hellhake, D., Bogner, J., Schmid, T., & Wagner, “Towards using coupling measures to guide black-box integration testing in component-based systems.” 2022.

[373] Q. Wang, Y. P. Shen, and Y. W. Chen, “Rule extraction from support vector machines,” *Guofang Keji Daxue Xuebao/Journal Natl. Univ. Def. Technol.*,

vol. 28, no. 2, pp. 106–110, 2006, doi: 10.1007/3-540-28803-1_10.

[374] G. Vilone and L. Longo, “Classification of Explainable Artificial Intelligence Methods through Their Output Formats,” *Mach. Learn. Knowl. Extr.*, vol. 3, no. 3, pp. 615–661, 2021, doi: 10.3390/make3030032.

[375] H. Strobelt, S. Gehrmann, H. Pfister, and A. M. Rush, “LSTMVis: A Tool for Visual Analysis of Hidden State Dynamics in Recurrent Neural Networks,” *IEEE Trans. Vis. Comput. Graph.*, vol. 24, no. 1, pp. 667–676, 2018, doi: 10.1109/TVCG.2017.2744158.

[376] L. He, N. Aouf, and B. Song, “Explainable Deep Reinforcement Learning for UAV autonomous path planning,” *Aerosp. Sci. Technol.*, vol. 118, p. 107052, 2021, doi: 10.1016/j.ast.2021.107052.