

UNIVERSIDADE FEDERAL DE MINAS GERAIS
Instituto de Ciências Exatas
Programa de Pós-Graduação em Ciência da Computação

Alisson Renan Svaigen

Design of Protection Mechanisms for the Internet of Drones

Belo Horizonte
2023

Alisson Renan Svaigen

Design of Protection Mechanisms for the Internet of Drones

Final Version

Dissertation presented to the Graduate Program in Computer Science of the Federal University of Minas Gerais in partial fulfillment of the requirements for the degree of Doctor in Computer Science.

Advisor: Antonio Alfredo Ferreira Loureiro
Co-Advisor: Linnyer Beatrys Ruiz Aylon

Belo Horizonte
2023

2023, Alisson Renan Svaigen.
Todos os direitos reservados

Svaigen, Alisson Renan.

S969d Design of protection mechanisms for the internet of drones
[recurso eletrônico] / Alisson Renan Svaigen – 2023.
1 recurso online (221 f. il, color.) : pdf.

Orientador: Antônio Alfredo Ferreira Loureiro.
Coorientador: Linnyer Beatrys Ruiz Aylon.
Tese (Doutorado) - Universidade Federal de Minas
Gerais, Instituto de Ciências Exatas, Departamento de
Ciências da Computação.
Referências: f. 203-221

1. Computação – Teses. 2. Aeronave não tripulada – Teses.
3. Internet - Medidas de segurança. 4. Direito a privacidade –
Teses. I. Loureiro, Antônio Alfredo Ferreira. II. Aylon, Linnyer
Beatrys Ruiz. III. Universidade Federal de Minas Gerais,
Instituto de Ciências Exatas, Departamento de Computação.
IV. Título.

CDU 519.6*82.9(043)

Ficha catalográfica elaborada pela bibliotecária Irenquer Vismeg Lucas Cruz
CRB 6/819 - Universidade Federal de Minas Gerais - ICEX



UNIVERSIDADE FEDERAL DE MINAS GERAIS
INSTITUTO DE CIÊNCIAS EXATAS
PROGRAMA DE PÓS-GRADUAÇÃO EM CIÊNCIA DA COMPUTAÇÃO

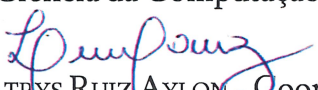
FOLHA DE APROVAÇÃO

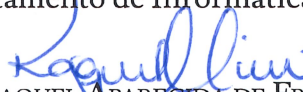
Design of Protection Mechanisms for the Internet of Drones

ALISSON RENAN SVAIGEN

Tese defendida e aprovada pela banca examinadora constituída pelos Senhores(a):



PROF. ANTONIO ALFREDO FERREIRA LOUREIRO - Orientador
Departamento de Ciência da Computação - UFMG


PROFA. LINNYER BEATRYS RUIZ AYLON - Coorientadora
Departamento de Informática - UEM


PROFA. RAQUEL APARECIDA DE FREITAS MINI
Ericsson Research


PROF. EDUARDO COELHO CERQUEIRA
Instituto de Tecnologia - UFPA


PROF. LUIZ FILIPE MENEZES VIEIRA
Departamento de Ciência da Computação - UFMG


PROF. HEITOR SOARES RAMOS FILHO
Departamento de Ciência da Computação - UFMG

Belo Horizonte, 20 de outubro de 2023.

*For Laercio and Marinete, my parents and the first professors
I had in my life.*

Acknowledgments

Initially, I thank the Creator for giving humanity the gift of free will. Through our personal choices, we can take our own paths, and therefore, taste the flavor of the flaw. It is only through the flaws that we can analyze our weaknesses to get stronger and develop solid works, contributing in a noble way to the progress of society.

I am greatly in debt to my advisor, Professor Antonio Loureiro, and to my co-advisor, Professor Linnyer Ruiz. I deeply thank you for the hours of guidance, always carried out in a proper and careful way, providing excellent support towards the addressing of this work in its excellence. Furthermore, I thank Professor Azzedine Boukerche for all the support during my stay at the University of Ottawa.

Special thanks to Lailla Milainny Siqueira Bine, my beloved fiancée, for her support, patience, attention, and countless tips and suggestions. Undoubtedly, without her presence, this journey would be much longer, tortuous, and difficult. We will be always together, in good and not-so-good moments, always supporting each other.

I sincerely thank the graduation staff from both UFMG and the University of Ottawa, and my graduate colleagues from Manna, Wisemap, and PARADISE Research Labs. Especially, Roniel and Katharinne Soares: Thank you so much for receiving Lailla and me in Canada! We will never forget all that you made for us!

Special thanks and acknowledgments to Debora Sandi, João Luiz Ramalheira, Wuigor Bine, Bianca Moggio, Felipe Mulhall, Marco Aurelio Paulino, and André Menegazzo. You are my friends for life, and I will always remember the sad days when you cheer me up. I would also thank to my relatives: my brother Willian Bruno, my sister-in-law Cristina, my nephews João Gabriel and Anna Livia, and my grandparents Nair e José.

I reserve for the last the most special thanks: to my parents. All the years of dedication and effort are the result not only of the study but also of parental encouragement and monitoring. Through them, my education began, and I was always determined to pursue knowledge, aiming to overcome all the difficulties imposed. Therefore, I end with this special thanks: thank you very much, Laercio and Marinete, for teaching me that there is no more dignified and noble journey other than the one of study and work.

This dissertation was partially supported by CAPES, CNPq, and FAPESP in Brazil, and partially supported by NSERC Funds and Canada Research Chairs Program, in Canada.

“We are at the very beginning of time for the human race. It is not unreasonable that we grapple with problems. But there are tens of thousands of years in the future. Our responsibility is to do what we can, learn what we can, improve the solutions, and pass them on.”

(Richard P. Feynman)

Resumo

Nos últimos anos, o uso de veículos aéreos não tripulados (VANTs) – conhecidos como drones – cresceu, evoluindo de serviços únicos para domínios colaborativos. Consequentemente, a Internet dos Drones (IoD) surgiu como um novo paradigma de rede móvel focado na integração de drones, coordenando o acesso ao espaço aéreo controlado e fornecendo um ambiente de comunicação robusto para nós aéreos e terrestres. Similar às redes móveis tradicionais, segurança e a privacidade são requisitos importantes a se garantir. IoD é um ambiente móvel com características particulares, diferindo dos paradigmas tradicionais. Portanto, os mecanismos de proteção existentes podem não ser adequados para garantir um nível de segurança e privacidade suficientes em IoD. Esses aspectos levam à duas importantes questões de pesquisa: (i) Os mecanismos de proteção atuais podem fornecer o mesmo nível de proteção para IoD quando comparados às redes móveis tradicionais? (ii) Se não podem, é possível adaptá-los para serem aplicados em IoD? Tendo em vista estes desafios, o objetivo principal desta tese é estudar o design de mecanismos de proteção para IoD considerando as suas características particulares. Além disso, realizamos um estudo aprofundado sobre os principais conceitos de IoD e sua relação com outras redes, quais as ameaças, quais os mecanismos de proteção existentes e como eles mitigam as ameaças. Este estudo revela a necessidade de aprimorar os mecanismos de proteção existentes para atender às características de IoD pois em maioria eles não podem oferecer o mesmo nível de proteção, ou mesmo não podem ser aplicados. Portanto, propomos um framework para orientar o design de mecanismos de proteção para IoD. Seguindo esse framework, avançamos no estado da arte em três frentes: (i) projetamos três novos Mecanismos de Proteção de Privacidade de Localização, também projetando um framework para aplicá-los cooperativamente; (ii) projetamos um mecanismo Anti-Jamming para IoD, sendo o primeiro mecanismo que mitiga os efeitos dos ataques de jamming em um ambiente com espaço aéreo restritamente definido; e (iii) introduzimos uma nova abordagem para identificar drones automaticamente no ambiente a partir de diferentes fontes, baseado no conceito de dissimilaridade. De modo geral, nossas contribuições aperfeiçoaram significativamente os níveis de segurança/privacidade de uma rede IoD comparado às soluções existentes, considerando um grupo de métricas relacionadas. Por fim, as contribuições deste estudo abrem espaço para o projeto de novos mecanismos de proteção sistematicamente, aumentando os níveis de segurança e privacidade em IoD.

Palavras-chave: Internet dos Drones; segurança; privacidade; mecanismos de proteção

Abstract

In the last years, the use of Unmanned Aerial Vehicles (UAVs), a.k.a. drones, has grown immensely, evolving from single-UAV services to collaborative domains. Hence, the Internet of Drones (IoD) emerged as a novel mobile network paradigm focused on the air-to-ground integration of drones, coordinating the access of drones to controlled airspace and providing a robust communication environment to the aerial and ground nodes. Likewise in traditional mobile networks, security and privacy are major requirements to be ensured. IoD is a unique mobile environment with particular characteristics, differing from the traditional paradigms. Hence, the existent protection mechanisms may not be adequate to ensure a proper level of security and privacy in IoD. These aspects lead to two important research questions: (i) Can the current protection mechanisms provide the same protection level to IoD when compared to the traditional mobile networks? (ii) If they can not, is it possible to adapt them to be applied in IoD? Bearing these challenges in mind, the main goal of this dissertation is to study the design of protection mechanisms for the IoD considering its particular characteristics. Apart from that, we conduct a thorough study regarding the main concepts of IoD and its relationship with other networks, what are the attacks that threaten this environment, what are the existent protection mechanisms, and how these mechanisms mitigate the attacks. This study reveals a need to enhance the existent protection mechanisms to meet the IoD characteristics since most of them can not offer the same protection level or even not be applied. Hence, we propose a framework to guide the design of IoD-based protection mechanisms. Following this framework, we advance the state of the art in three fronts: (i) we design three novel IoD-related Location Privacy Protection Mechanisms (LPPMs), also designing a framework to apply them cooperatively; (ii) we design an Anti-Jamming mechanism for IoD, being the first mechanism that mitigates the effects of Jamming Attacks in an aerial environment with restricted available airspace; (iii) and we introduce a new approach to automatically identify drones in the environment from different sources, based on the dissimilarity concept. Summarily, our contributions significantly enhanced the security/privacy levels of the IoD facing the existent solutions, considering a pool of related metrics. Finally, the contributions of this study make room for the design of novel protection mechanisms systematically, enhancing the IoD security and privacy levels.

Keywords: Internet of Drones; security; privacy; protection mechanisms

List of Figures

1.1	Related research field and dissertation's contribution	25
2.1	Action areas of UAV networks	28
2.2	IoD layered network architecture	30
2.3	Venn diagram of IoD privacy issues and their relation with the IoD components	34
3.1	IoD-related attacks taxonomy	38
3.2	An example of EA being performed	42
3.3	An example of SA being performed	45
3.4	An example of HA being performed	47
3.5	Framework to guide the design of IoD protection mechanisms	63
4.1	Example of a MZ region	69
4.2	Comparison of a traditional MZ and t-MixDrones proposal	72
4.3	Architecture of t-MixDrones	73
4.4	Conception of helical mobility to avoid collision between two drones	75
4.5	Simulated urban scenario for t-MixDrones performance evaluation	85
4.6	Results of Coverage Rate regarding the t-MixDrones performance evaluation .	88
4.7	Results of Re-anonymization Average Rate regarding the t-MixDrones performance evaluation	89
4.8	Results of Trajectory Matching Accuracy regarding the t-MixDrones performance evaluation	90
4.9	Results of Increasing Travel Time Rate regarding the t-MixDrones performance evaluation	92
4.10	MixRide concept	97
4.11	Results of the MixRide performance evaluation	102
4.12	Topology-based dummy generation concept	109
4.13	Network traffic comparison regarding TDG performance evaluation	112
4.14	Location privacy results regarding TDG performance evaluation	113
4.15	IoDAPM concept	118
4.16	Simulated environment regarding IoDAPM performance evaluation	123
4.17	Average Quality of Service (QoS) per aerial zone in the training phase	125
4.18	Cumulative rate of observed conditions per episode	126
4.19	Results of each considered metric of the comparative evaluation	127
4.20	Concept of the designed Remote ID Location-based Attack	131

4.21	Example of trajectory analysis comparing the actual drone's trajectory and the tracked by the Remote ID-based attack	135
4.22	Case study results of the evaluated metrics for each combination of number of eavesdroppers \times protocol	138
5.1	IoD environment as an ITS	144
5.2	Example of how JA affect drone trajectory in IoD	145
5.3	The proposed IoD-JAPM workflow	146
5.4	Delimitation of unused aerial space between two parallel airways with different altitudes	148
5.5	Airway analysis step of IoD-JAPM performed by a drone	149
5.6	The typical Hazard Region calculated in each topology over all the airways . .	162
5.7	Results of Hazard Region Rate regarding IoD-JAPM performance evaluation .	163
5.8	Results of Drones with Affected Path Planning Rate regarding IoD-JAPM performance evaluation	164
5.9	Results of Increasing Flight Distance Rate regarding IoD-JAPM performance evaluation	165
5.10	Results of Increasing Power Consumption Rate regarding IoD-JAPM performance evaluation	165
6.1	Proposed rhythm-based ADD methodology	176
6.2	Dissimilarity representation workflow	187
6.3	DissIdent approach: Deployment Phase	188
6.4	Confusion Matrix regarding the results of supervised-based approaches for UAV identification	194
6.5	Confusion Matrix regarding the results of DissIdent UAV identification for each configuration	194
6.6	Confusion Matrix regarding the results of the Clustering approaches	196
6.7	Results of clusters distribution for each clustering technique	197

List of Tables

2.1	Comparison between traditional mobile networks and the IoD	32
2.2	IoD application aspects and what privacy issues they impact	35
3.1	Main aspects of the IoD-based attacks	39
3.2	IoD elements that demand protection and their relationships	51
3.3	Protection mechanisms and what element groups they protect	53
3.4	Protection mechanisms and what attack categories they can mitigate	54
4.1	Recent studies of MZ mechanism	70
4.2	Relation of Factors Between MZs for VANETs and for IoD	71
4.3	Mapping of the main concepts of ACO and BioMixD	80
4.4	Simulation parameters regarding the t-MixDrones performance evaluation	87
4.5	Related Studies of Aerial and Terrestrial Transportation Collaboration	96
4.6	Simulation Parameters regarding the MixRide performance evaluation	101
4.7	Recent studies related to dummy-based LPPMs	107
4.8	Simulation parameters regarding the TDG performance evaluation	111
4.9	Categorization of the IoD environment conditions	116
4.10	IoDAPM Setup	124
4.11	Remote ID Case Study: Simulation Parameters	136
5.1	Topology Attributes	159
5.2	Simulation Parameters	160
6.1	Early and late fusion configurations of the proposed rhythm-based strategy for ADD	178
6.2	Folds distribution regarding training and testing of the proposed rhythm-based strategy for ADD	179
6.3	Final configuration of parameter tuning and the corresponding accuracy of the proposed rhythm-based strategy for ADD	180
6.4	Binary classification results for balanced and unbalanced data regarding the performance evaluation of the proposed rhythm-based strategy for ADD	181
6.5	Best confusion matrix of binary classification for balanced classes regarding the performance evaluation of the proposed rhythm-based strategy for ADD (LF1 – max rule).	182

6.6	Best confusion matrix of binary classification with unbalanced classes regarding the performance evaluation of the proposed rhythm-based strategy for ADD (LF1 – max rule).	183
6.7	Multiclass classification results for the balanced and unbalanced datasets regarding the performance evaluation of the proposed rhythm-based strategy for ADD	183
6.8	Confusion matrix for multiclass classification with balanced classes regarding the performance evaluation of the proposed rhythm-based strategy for ADD (LF3 – product rule).	184
6.9	Confusion matrix for multiclass classification of unbalanced data regarding the performance evaluation of the proposed rhythm-based strategy for ADD (LF3 – max rule).	184
6.10	Comparison of the baseline and our proposal – Best accuracy rate for binary and multiclass classification in balanced data scenarios.	185
6.11	Comparison of accuracy and F1-score between the baseline and our proposal. .	185
6.12	Summarization of the results regarding the accuracy obtained by each approach	193

List of Symbols

Δ_t	Time interval
λ	Communication channel
\mathcal{AT}	set of actions to take by a network agent
\mathcal{AZ}	Set of aerial zones
\mathcal{A}	Set of attacks/attackers/adversaries
\mathcal{CT}	Set of cyber tools
\mathcal{C}	Set of environmental conditions
\mathcal{DH}	Set of drone-related hardwares
\mathcal{DS}	Set of drone services
\mathcal{D}	Set of drones
\mathcal{FV}	set of feature vectors
\mathcal{GV}	Set of ground vehicles
\mathcal{MN}_{IoD}	IoD Mobile Network
\mathcal{MN}_T	Traditional Mobile Network
\mathcal{N}	Set of network nodes
\mathcal{PM}	Set of protection mechanisms
\mathcal{Q}	Set of location-based queries
\mathcal{SI}	set of signal inputs
\mathcal{S}	set of network state
\mathcal{U}	Set of IoD 3 rd -party users
\mathcal{V}	set of vertiports
\mathcal{Z}	Set of management nodes

ε Energy level

$G = (V, E)$ Airspace graph model

kl Network knowledge level

r Coverage radius

CS Coordinate System

M Set of metrics

,

List of Abbreviations

A2A Air-to-Air.	HAP High-Altitude Platform.
A2G Air-to-Ground.	IoD Internet of Drones.
ACO Ant Colony Optimization.	IoFT Internet of Flying Things.
ADD Automatic Drone Detection.	IoT Internet of Things.
ATC Air Traffic Control.	ITS Intelligent Transportation System.
B5G 5G and Beyond.	ITTR Increasing Travel Time Rate.
CNN Convolutional Neural Network.	JA Jamming Attack.
CRNN Convolution Recurrent Neural Networks.	LBS Location-based Server.
D2D Drone-to-drone.	LBSN Location-based Services Network.
D2G Drone-to-Ground.	LoS Line of Sight.
D2I Drone-to-infrastructure.	LPA Location Privacy Attack.
D2X Drone-to-Everything.	LPL Location Privacy Leakage.
DAA De-Anonymization Attack.	LPPM Location Privacy Protection Mechanism.
DaaS Drone as a service.	LSTM Long Short-Term Memory.
DNN Deep Neural Network.	MiMA Man-in-the-middle Attack.
DOA Direction of Arrival.	ML Machine Learning.
DoS Denial of Service.	MZ Mix Zones.
DRTP Distance Rate per Trajectory Point.	MZP Mix Zones Placement.
DTW Dynamic Time Warping.	PCR Power Consumption Rate.
EA Eavesdropping Attack.	PM Protection Mechanism.
FAA Federal Aviation Administration of the United States.	PoI Points of Interest.
FANET Flying Ad hoc Network.	QoS Quality of Service.
G2G Ground-to-Ground.	RDLDA Real Drone Location Discovery Accuracy.
GNSS Global Navigation Satellite System.	RF Radio Frequency.
GPR Generated Packets Rate.	RL Reinforcement Learning.
GPS-SA GPS Spoofing Attack.	RNN Recurrent Neural Networks.
HA Hijacking Attack.	RSU Road-Side Unit.

RTTDR Real Trajectory Time Discovery Rate.
SA Spoofing Attack.
SDN Software Defined Network.
SE/HE Software / Hardware Exploitation.
SVM Support Vector Machine.
SWaP Size, Weight and Power.
TAA Traffic Analysis Attack.
TDG Topology-based Dummy Generation.
TDR Trip Delay Rate.
TMA Trajectory Matching Accuracy.
TTP Trusted Third Party.
UAV Unmanned Aerial Vehicle.
UWSN Underwater Wireless Sensor Networks.
VANET Vehicular Ad hoc Network.
WSN Wireless Sensor Networks.
ZSP Zone Service Provider.

,

List of Publications

Referred Journal Papers

1. Svaigen, A. R., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. (2022). BioMixD: A Bio-inspired and Traffic-aware Mix Zone Placement Strategy for Location Privacy on the Internet of Drones. **Computer Communications**. Vol. 195. pp. 111–123.
2. Svaigen, A. R., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. (2022). Design Guidelines of the Internet of Drones Location Privacy Protocols. **IEEE Internet of Things Magazine**. vol. 5, no. 2, pp. 175-180, June.
3. Svaigen, A. R., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. (2023). Trajectory Matters: Impact of Jamming Attacks Over the Drone Path Planning on the Internet of Drones. **Ad Hoc Networks**, 146, 103179.
4. Svaigen, A. R., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. (2023). Security in the Industrial Internet of Drones. **IEEE Internet of Things Magazine**. vol. 6, no. 3, pp. 110-116, September.
5. Svaigen, A. R., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. Attacks in the Internet of Drones through a Privacy Perspective: Models, Trendings, and Challenges. **ACM Transaction on Internet of Things**. (*First Round of reviews*).

Referred Conference Papers

1. Svaigen, A. R., Ramos H. S., Ruiz L. B., and Loureiro, A. A. F. (2019). Dynamic Temporal Mix-zone Placement Approach for Location-based Services Privacy. In **2019 IEEE Latin-American Conference on Communications (LATIN-COM)**, pages 1–6.
2. Svaigen, A. R., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. (2021). Mix-Drones: A Mix Zones-based Location Privacy Protection Mechanism for the Internet of Drones. In **24th International ACM Conference on Modeling, Analysis and Simulation of Wireless and Mobile Systems**, pages 181–188.

3. Svaigen, A. R., Bine, L. M. S., Pappa, G. L., Ruiz, L. B., and Loureiro, A. A. F. (2021). Automatic Drone Identification Through Rhythm-based Features for the Internet of Drones. In **2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)**, pages 1417–1421.
4. Svaigen, A. R., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. (2022). A Topological Dummy-based Location Privacy Protection Mechanism for the Internet of Drones. In **ICC 2022-IEEE International Conference on Communications**, pages 3735–3740.
5. Svaigen, A. R., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. (2022). Um Mecanismo de Proteção Ciente de Vias Aéreas Contra Jamming Attacks para a Internet dos Drones. In **Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos**, pages 405-418. Porto Alegre: SBC.
6. Svaigen, A. R., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. (2022). Is the Remote ID a Threat to the Drone’s Location Privacy on the Internet of Drones? In **20th ACM International Symposium on Mobility Management and Wireless Access (MobiWac ’22)**. Association for Computing Machinery, New York, NY, USA, 81–88.
7. Svaigen, A. R., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. (2022). Analyzing the UAVs Traffic Flow to Enhance the Drone’s Anonymization on the Internet of Drones. In **12th ACM International Symposium on Design and Analysis of Intelligent Vehicular Networks and Applications (DIVANet ’22)**. Association for Computing Machinery, New York, NY, USA, 45–52.
8. Svaigen, A. R., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. (2022). MixRide: An energy-aware location privacy protection mechanism for the internet of drones. In **GLOBECOM 2022-2022 IEEE Global Communications Conference**, pages 3527-3532. IEEE.
9. A. R. Svaigen, A. Boukerche, L. B. Ruiz and A. A. F. Loureiro, ”DissIdent: A Dissimilarity-based Approach for Improving the Identification of Unknown UAVs,” **2023 IEEE 34th Annual International Symposium on Personal, Indoor and Mobile Radio Communications (PIMRC)**, Toronto, ON, Canada, 2023, pp. 1-6.
10. Svaigen, A. R., Boukerche, A., Ruiz, L. B., and Loureiro, A. A. F. (2023). IoDAPM: A Reinforcement Learning Approach for Dynamic Assignment of Protection Mechanisms in IoD. In **Int’l ACM Conference on Modeling Analysis and Simulation of Wireless and Mobile Systems (MSWiM ’23)**. Association for Computing Machinery, New York, NY, USA, 147–154.

Contents

1	Introduction	22
1.1	Problem Statement	23
1.2	Goals and Contributions	24
1.3	Dissertation Outline	26
2	Basic Concepts of the Internet of Drones (IoD)	27
2.1	UAV Networks: Where We Are and Where We Will Go	27
2.2	IoD as a Layered Mobile Network	29
2.3	The Relation Between IoD and Traditional Mobile Networks	32
2.4	Concepts of Security and Privacy in the IoD	33
2.5	Chapter Remarks	36
3	Attacks and Protection Mechanisms in IoD: Overview and Design Guidelines	37
3.1	Attacks in IoD	37
3.1.1	De-Anonymization Attack (DAA)	39
3.1.2	Traffic Analysis Attack (TAA)	41
3.1.3	Eavesdropping Attack (EA)	41
3.1.4	Jamming Attack (JA)	43
3.1.5	Spoofing Attack (SA)	45
3.1.6	Hijacking Attack (HA)	46
3.1.7	Software and Hardware Exploitation (SE/HE)	48
3.2	Protection Mechanisms for IoD: Do We Need to Reinvent the Wheel?	49
3.2.1	Elements to Protect in IoD	50
3.2.2	Location Privacy Protection Mechanisms	53
3.2.3	Anti-Jamming Mechanisms	55
3.2.4	Automatic Drone Detection	56
3.2.5	Anti-Spoofing Mechanisms	57
3.2.6	Cryptographic-based Mechanisms	58
3.2.7	Air Traffic Regulations	58
3.3	Trends and Challenges	60
3.4	A Framework to Guide the Design of IoD-based Protection Mechanisms	61
3.5	Chapter Remarks	63

4	Design of Location Privacy Protection Mechanisms for IoD	65
4.1	Introduction	66
4.2	t-MixDrones	67
4.2.1	Threat Model	67
4.2.2	Mix Zones Concepts	68
4.2.3	Related Studies	70
4.2.4	t-MixDrones Architecture	72
4.2.5	MixDrones Module	74
4.2.6	BioMixD Module	77
4.2.7	Simulation Setup and Performance Evaluation	84
4.2.8	Results and Discussion	86
4.3	MixRide	93
4.3.1	Threat Model	94
4.3.2	Related Studies	95
4.3.3	Design of MixRide	95
4.3.4	Simulation Setup and Performance Evaluation	100
4.3.5	Results and Discussion	101
4.4	TDG	105
4.4.1	Threat Model	105
4.4.2	Related Studies	107
4.4.3	Design of TDG	108
4.4.4	Simulation Setup and Performance Evaluation	110
4.4.5	Results and Discussion	112
4.5	IoDAPM: An RL Approach for Dynamic Assignment of LPPMs	115
4.5.1	Fundamental Aspects	116
4.5.2	Design of IoDAPM	117
4.5.3	Simulation Setup and Performance Evaluation	123
4.5.4	Results and Discussion	125
4.6	Case Study: Impact of Remote ID Rule in the Drone's Location Privacy	128
4.6.1	Related Proposals	130
4.6.2	Design of Remote ID Location-based Attack	130
4.6.3	Simulation Setup and Performance Evaluation	134
4.6.4	Results and Discussion	137
4.7	Chapter Remarks	140
5	Design of Anti-Jamming Mechanisms for IoD	142
5.1	Introduction	142
5.2	Application Scenario and Threat Model	143
5.3	Related Studies	144

5.4	Design of IoD-JAPM	147
5.4.1	Airway Analysis	148
5.4.2	Hazard Region	152
5.4.3	Path Planning Reformulation	154
5.5	Simulation Setup and Performance Evaluation	157
5.5.1	Compared Approaches	157
5.5.2	Simulation Parameters	158
5.5.3	Metrics	159
5.6	Results and Analysis	161
5.7	Chapter Remarks	168
6	Design of Automatic Drone Detection Strategies for IoD	170
6.1	Introduction	170
6.2	Application Scenario and Threat Model	172
6.3	Related Studies	173
6.4	Design of a Smart Rhythm-Based Strategy for ADD	175
6.4.1	Feature Extraction	175
6.4.2	Classification Model and Multimodality Definitions	177
6.4.3	Setup and Performance Evaluation	178
6.4.4	Results and Discussion	181
6.5	Design of DissIdent	185
6.5.1	Dissimilarity Representation	186
6.5.2	DissIdent Deployment Phase	188
6.5.3	Monitoring Phase	189
6.5.4	Simulation Setup and Performance Evaluation	191
6.5.5	Results and Discussion	193
6.6	Chapter Remarks	198
6.6.1	New Research Directions for ADD	198
7	Conclusion and Future Work	200
7.1	Concluding Remarks	200
7.2	Research Directions	202
	References	204

Chapter 1

Introduction

Over the last years, the Unmanned Aerial Vehicle (UAV), also known as “drone”, has gained new business interests in different fields. For instance, urban centers, rural areas, industry, and ports [1, 2]. Several companies have been exploring Drone as a service (DaaS). For instance, Amazon Prime Air is an on-demand package delivery service. DroneDeploy¹ is a company that provides drone-based solutions for agriculture. Indeed, the Federal Aviation Administration of the United States (FAA) reported the register of 869,472 drones until July of 2023 in which 59% were for recreational purposes and 41% for commercial operation [3]. Several reports point out a huge growth of the drone market, resulting in an increase of about U\$ 35 billion by 2025 [4].

From a Computer Network point-of-view, the widespread utilization of drone-based technology in the civilian context creates a heterogeneous mobile network environment composed of different drones with different purposes acting as nodes. Drones can significantly enhance Intelligent Transportation System (ITS) due to drone mobility, autonomous operation, and communication capabilities [1]. Given this growth, the scope of UAV networks have evolved from single to multi-UAV applications. Therefore, the next generation of UAV networks will require a robust, reliable, and inter-operable network to accomplish organized and collision-avoidance airspace.

Anticipating this environment, Gharibi et al. [5] proposed a layered network architecture, named the Internet of Drones (IoD). Although the architecture name indicates a general purpose, it aims to coordinate the access of UAVs to controlled airspace, providing navigation services to the drones through a Zone Service Provider (ZSP), which acts as a base station. A remarkable characteristic of the network is the presence of airways, referring to delimited aerial regions where drones can fly, being similar to the roadways concept. Despite this definition, IoD can also refer to UAV-networks in broad scope, ranging from swarm-based to an interoperable environment, integrating different network infrastructures.

Compared to ground mobile networks, such as Vehicular Ad hoc Network (VANET), IoD has particular characteristics, for instance: drones move fast over the airspace limited by the airways; they communicate at the Line of Sight (LoS); and have Size, Weight and

¹<https://www.dronedeploy.com/solutions/agriculture/>

Power (SWaP) limitations [1], which can affect the provided Quality of Service (QoS) deeply. Thus, IoD demands a thorough investigation in the design of network protocols since its characteristics can affect the performance of an existing one [6].

1.1 Problem Statement

Security is one of the major challenges to be addressed in this novel and unique scenario [1, 4, 7, 8, 9]. Since drones are in the airspace, there is a range of attack methods that can be more harmful when compared to grounded mobile networks [1, 7]. Likewise, privacy is a concept interlaced with mobile network security. It is commonly defined as the protection level that needs to be addressed regarding the user's personal information [1, 10]. Nonetheless, considering the IoD context, the privacy concept can be extended, incorporating a great number of environmental aspects, such as the drone's privacy as a device, the communication channel's privacy, and the territorial aerial privacy.

Likewise in other mobile networks, both the network infrastructure and the surrounding environment must have Protection Mechanisms (PMs) to ensure well-known security properties (e.g., availability, integrity, and confidentiality) [4, 8] not just for drones but also for other elements of the environment. For instance, ground vehicles must communicate with drones through a reliable channel. Buildings and people must be aware of the presence of drones and be protected from a potential malfunction [1, 5, 7].

On one hand, PMs have been investigated in other mobile paradigms, such as VANETs, IoT [11], and general UAV-related networks [8]. On the other hand, they have not been applied in IoD or even verified for the threats that can affect this environment. Indeed, this issue is a serious challenge since this prospected environment represents a novel scenario with particular characteristics [1]. Thus, the design of PMs related specifically with IoD is in its initial steps.

Therefore, some research questions arise. Let us consider a set of well-known attacks \mathcal{A} , and a set of protection mechanisms \mathcal{PM} applied as countermeasures in a traditional mobile network environment \mathcal{MN}_T . Also, let us consider an IoD environment \mathcal{MN}_{IoD} , subjected to different attacks from \mathcal{A} .

- ▶ **RQ1:** Can the protection mechanisms of \mathcal{PM} provide the same protection level to \mathcal{MN}_{IoD} when compared to \mathcal{MN}_T ?
- ▶ **RQ2:** If **RQ1** is false, is it possible to adapt a protection mechanism $\rho \in \mathcal{PM}$, aiming to enhance the protection level provided to \mathcal{I} ?

These questions represent opened challenges in the Computer Networks research field, specifically, in the security and privacy aspects of IoD. Obtaining the answers to these questions can bring significant advancements regarding the deployment of PMs in real-world IoD environments, contributing to the planning and development of this prospected scenario.

1.2 Goals and Contributions

The **main goal** of this dissertation is **to study the design of Protection Mechanisms (PMs) for the Internet of Drones (IoD) paradigm, considering the particular characteristics of this environment**. To address this goal, we conduct a thorough study regarding the main concepts of IoD and its relationship with other networks, what are the attacks that threaten this environment, what are the existent PMs, and how these mechanisms mitigate the attacks.

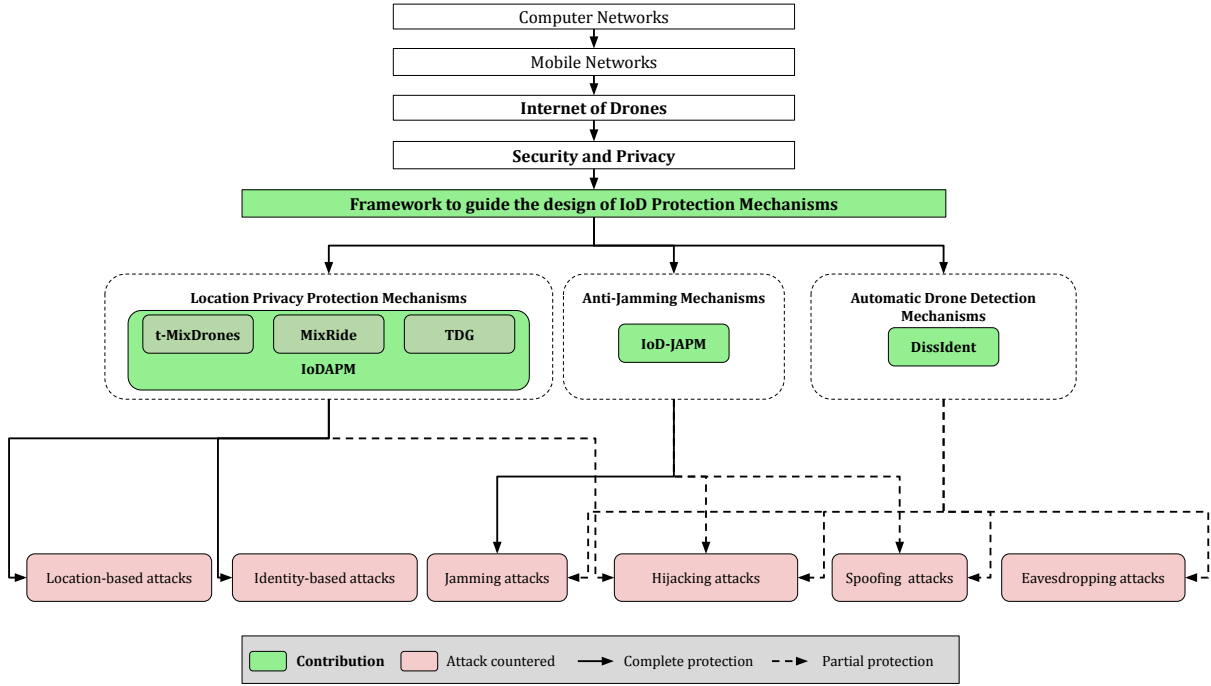
This study reveals seven major attacks that can affect severely IoD due to its particular characteristics: Traffic Analysis Attack (TAA); Software/Hardware Exploitation (SE/HE); De-Anonymization Attack (DAA); Jamming Attack (JA); Eavesdropping Attack (EA); Spoofing Attack (SA); Hijacking Attack (HA). Likewise, we surveyed five major PMs that can mitigate the occurrence of these attacks: Location Privacy Protection Mechanism (LPPM); Automatic Drone Detection (ADD); Anti-Jamming mechanisms; Anti-Spoofing; and Cryptographic-based mechanisms. From that, there is a need to investigate whether the PMs meet the IoD characteristics and whether they can offer the same protection level or even be applied.

From this study, we advance the state-of-the-art in the IoD security/privacy field on four different fronts. Figure 1.1 summarizes the main contributions of this dissertation. We describe them as follows.

- ▶ **Guidelines Framework:** we propose a framework to guide the design of IoD-related PMs, systematically. This framework encompasses the definition of the IoD scenario, the attack modeling, the PMs modeling, the conduction of experiments, and their evaluation. The design of all the subsequent PMs follows this framework;

- ▶ **Location Privacy Protection Mechanisms (LPPMs):** this front represents a major contribution of this dissertation since LPPMs have not been investigated in the IoD environment. Our contributions enhanced the level of provided location privacy facing

Figure 1.1: Related research field and dissertation's contribution



Source: Elaborated by the author

both location and identity-based attacks, also mitigating the occurrence of Hijacking Attack (HA). The contributions involve the design of the following novel mechanisms:

- **t-MixDrones**: a Mix Zones-based (MZ) Location Privacy Protection Mechanism (LPPM) for the IoD. This mechanism overcomes the location privacy protection level when compared to traditional MZ-based approaches, considering IoD scenarios with dense topology;
- **MixRide**: an energy-aware LPPM for the IoD. This mechanism handles the energy constraints of the drones, providing location privacy through the aerial-grounded vehicle collaboration, where the drones take a ride with grounded vehicles, changing their pseudonyms while saving energy;
- **TDG**: a topological dummy-based LPPM for the IoD. We propose TDG for scenarios with sparse topology, overcoming the application that lacks the two former mechanisms.

The three proposed LPPMs have better performance in different environment conditions. Therefore, we also design a Reinforcement Learning (RL) approach for the dynamic assignment of PMs in IoD, named IoDAPM. The approach aims to improve the QoS provided by the network from a transition model of rewards, obtained by previous mechanism assignments made in the network, considering the environmental conditions.

- ▶ **Anti-Jamming mechanism:** we shed light on the impact of the JA in the IoD, mainly regarding the drone path planning and, therefore, the drone trajectory. To overcome these challenges, we propose the **IoD-JAPM**, an airway-aware protection mechanism against JA, ranging from analyzing the airway's availability to the potential reformulation of the drone path planning. IoD-JAPM protects directly against JA, also mitigating the occurrence of both HA and Spoofing Attack (SA);

- ▶ **Automatic Drone Detection (ADD) strategies:** we introduce new approaches to detect drones in the airspace. Our contributions in this field are twofold. First, we investigate the drone's propeller acoustic signal as a primary source to automatically detect and identify drones, demonstrating that this source provides suitable detection/identification rates when combined with Machine Learning (ML) techniques. Also, we introduce the dissimilarity concept to detect unknown drones in the airspace through the **DissIdent** mechanism. This solution can identify patterns from different features through a smart workflow involving ML and clustering concepts. Considering that our findings represent a new front of research, we discuss different techniques that can be applied to enhance our proposed approaches.

1.3 Dissertation Outline

The remainder of this dissertation is organized into six chapters, described as follows. Chapter 2 presents the fundamental concepts of the IoD and security/privacy aspects. Chapter 3 brings an overview of attacks that threaten this environment, and the existent protection mechanisms facing these attacks. This chapter also proposes a framework to guide the design of novel IoD-related PMs. Chapter 4 presents the contributions related to the design of novel LPPMs. Chapter 5 brings the contribution related to Anti-Jamming mechanisms, named IoD-JAPM. Chapter 6 presents the contributions regarding the design of ADD strategies. Last but not least, Chapter 7 summarizes all the contributions addressed in this dissertation. Additionally, this chapter lists new challenges to tackle, shedding light on new research directions regarding the IoD security and privacy fields.

Chapter 2

Basic Concepts of the Internet of Drones (IoD)

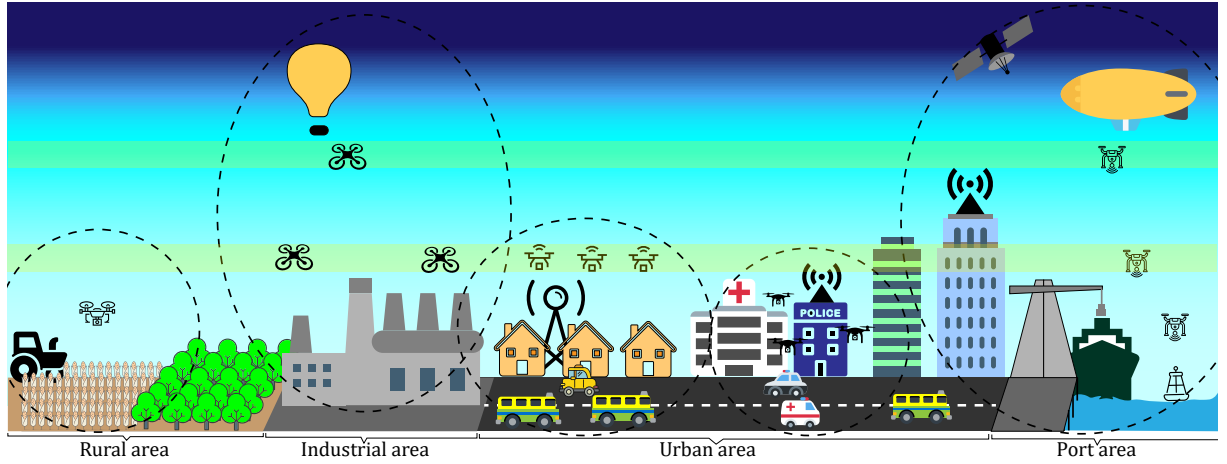
This chapter presents the fundamental concepts regarding IoD. Initially, we present an overview of the current scenario regarding UAV Networks, and the prospection for the next generation of this paradigm (Section 2.1). After, we state the basic concepts of the IoD considering it as a layered mobile network [5] (Section 2.2). Section 2.3 brings the relationship between IoD and traditional mobile networks. After, in Section 2.4, we present security and privacy aspects involved with IoD. Lastly, we state our chapter remarks (Section 2.5).

2.1 UAV Networks: Where We Are and Where We Will Go

Over the years, UAV-related applications changed from single to multi-UAV services [12]. They have boosted the usage of Drone as a service (DaaS) in different market fields, including urban centers, rural areas, industry, and also oceanic environments. These applications have the potential to bring a new panorama of UAV networks, improving the people's quality of life in personal and social aspects [1, 6].

Figure 2.1 illustrates different areas where UAV networks can act. A single drone can support rural activities by taking aerial images of crops and transmitting them to cloud systems to process these data by applying AI-based techniques and optimizing rural tasks[13]. UAVs can also communicate with High-Altitude Platform (HAP) (e.g., atmospheric balloons), creating a collaborative aerial network to assist ground operators, such as in the industry [14]. In this direction, UAVs can collaborate with well-established ground systems such as vehicular networks. For instance, drones can assist police in surveillance services [2]. Sensor networks (e.g. Underwater Wireless Sensor Networks

Figure 2.1: Action areas of UAV networks



Source: Elaborated by the author

(UWSN)) can also collaborate with UAV networks, where drones can move fast through the ocean surface and collect data fast [15].

In the near future, there is a prospect that these services will occur simultaneously, leading to an enhanced Intelligent Transportation System (ITS), empowering an environment where drones will fly over our heads, performing several services in different areas and from different companies [16]. Therefore, this environment requires a robust, reliable, and interoperable network to manage the airspace traffic flow (similar to terrestrial roads) and also to provide a fair and shareable communication channel.

The emerging of new communication protocols and technologies, mainly the 5G and Beyond (B5G), leverage drones to a higher level in a such way that drones can play a crucial role in assisted networks, or even represent a novel mobile network paradigm [2, 12]. Several researchers investigated drone technology acting as a well-structured mobile network. These studies lead to the concepts of different paradigms, such as Internet of Flying Things (IoFT) [17] and Flying Ad hoc Network (FANET) [18]. Most of them focus on communication issues, neglecting the flight policies in aerial space.

However, from an ITS point-of-view, it is indispensable to define properly how the involved nodes can move, and where they can move. Prospecting a UAV-based ITS neglecting their flight policies is similar to designing a VANET infrastructure without considering how and where the ground vehicles will move. Moreover, defining these aerial policies leads to an unprecedented environment, mainly regarding the security/privacy aspects, and how to ensure them.

Nonetheless, Gharibi et al. [5] proposed an UAV network model named Internet of Drones (IoD). This model covers most of the issues discussed above. Therefore, this dissertation takes IoD as the reference network model. The next section discusses this model in detail.

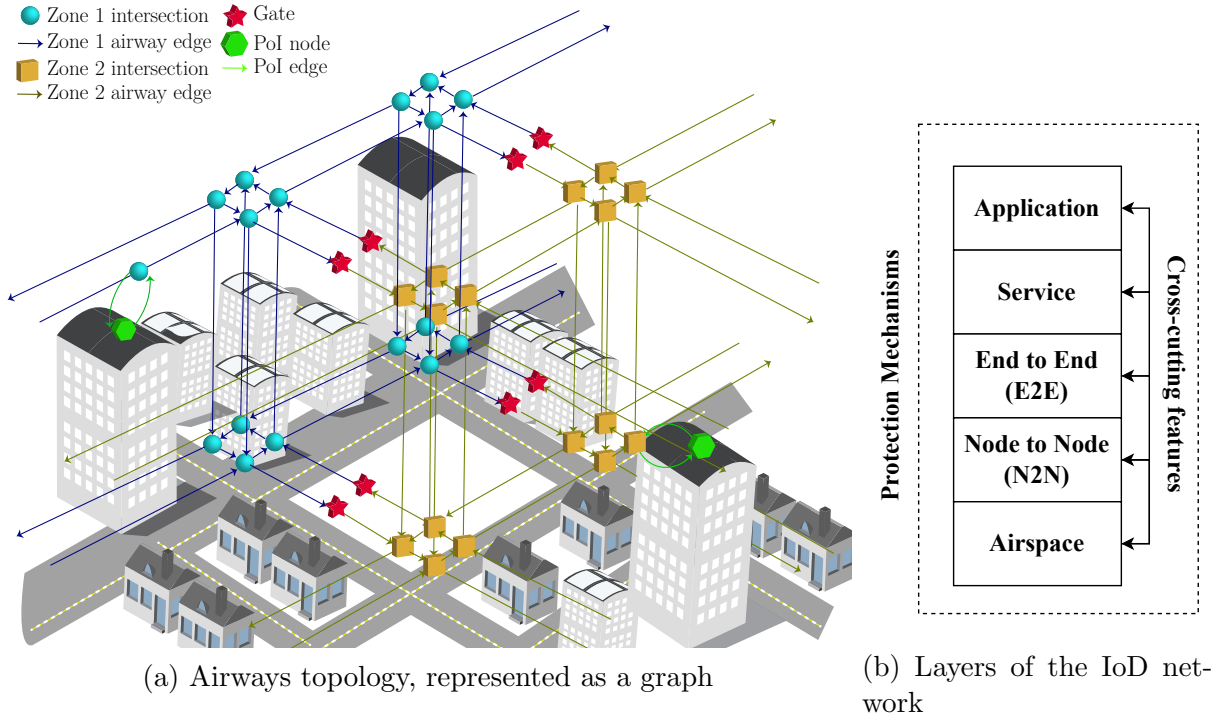
2.2 IoD as a Layered Mobile Network

Gharibi et al. [5] modeled IoD as a cross-layered network architecture to allow coordinated access to the airspace for drones. This architecture is based on three large-scale networks: Air Traffic Control (ATC), Cellular network, and the Internet. These networks achieve some of the goals or functionalities desired for the IoD, in which the model brings together the advantages of each one. In this dissertation, we consider the following definitions:

- **Internet of Drones (IoD):** it is any mobile network environment \mathcal{MN}_{IoD} that involves a set of nodes \mathcal{N} and a subset of drones \mathcal{D} , such that $\mathcal{D} \subset \mathcal{N}$. The network infrastructure related to \mathcal{MN}_{IoD} has as the main goal to provide a fair aerial space to \mathcal{D} , where they can perform a set of services \mathcal{DS} for different third-parties networks and their nodes (represented by the IoD users \mathcal{U}). The navigation over this aerial space is modeled as a graph $G = (V, E)$, being managed by a set of management nodes \mathcal{Z} , such that $\mathcal{Z} \subset \mathcal{N}$. In a nutshell, the proposed architecture can be described in terms of infrastructure and layers. The IoD infrastructure is composed of drones, Zone Service Provider (ZSP), and airways.
- **Drone:** it is a UAV node $d \in \mathcal{D}$, such that $\mathcal{D} \subset \mathcal{N}$, capable of operating in the aerial network \mathcal{MN}_{IoD} . A given drone d has specific configurations and requirements of hardware and software, in accordance with \mathcal{MN}_{IoD} .
- **Zone Service Provider (ZSP):** it is a base station node $z \in \mathcal{Z}$ such that $\mathcal{Z} \subset \mathcal{N}$, providing navigation information to \mathcal{D} considering a fairness policy. A given ZSP z communicates with a set of drones $\mathcal{D}_z \subset \mathcal{D}$ through a shared communication channel λ .
- **Airways:** in a computational point of view, the airways represent the edges E from a graph $G = (V, E)$, being the minimum flyable aerial distance between two nodes $v_1, v_2 \in V$ such that each node can represent an intersection point between airways, or a Points of Interest (PoI), such as a recharge station or even restricted airspace (e.g., airport region).

In a broader scope, each ZSP covers a geographic region, called a zone. However, two or more ZSPs may stay in the same zone. ZSP follows the governing laws regarding the airways, intersections, and nodes to ensure a safe and reliable drone traffic flow. Moreover, a ZSP must be independent of drone companies, and consequently, it can not have access to the application's contextual information unless in the cases that the drone allows it. To manage all this infrastructure, the network assumes a cloud system, connecting the ZSPs

Figure 2.2: IoD layered network architecture



(a) Airways topology, represented as a graph

(b) Layers of the IoD network

Source: Inspired by Gharibi et al. [5]

and, therefore, allowing communication with each other or with third-party entities over the cloud. It means that ZSPs provide not only navigation but the main communication channel for drones.

Figure 2.2a illustrates an example of the IoD infrastructure, where each sphere represents a node, and each arrow represents an edge. In this scenario, two similar parallel airways are placed over different altitudes. Considering that different ZSPs will manage the airspace synchronously, we note that nodes and edges are blue (representing Zone 1) on one side, whereas they are yellow on the other side, representing airspace coordinated by a different ZSP. The red stars represent the borders between two zones, named gates. As drones perform various services, they will have PoIs created in real-time, expressed through the green objects.

Furthermore, IoD has five layers to ensure flexibility, scalability, and maintainability of the network. Differing from the traditional Internet stack protocol, the layers are crossed by each other, where upper layers can access lower layers and not just the layers directly below them. Figure 2.2b illustrates how these layers are organized. Their expected features are presented as follows.

► *Airspace*: it is responsible for implementing the navigation directives between the ZSPs and the drones. The expected features covered by this layer are the airways' map representation, drone's location guidance & track, navigation vectors, collision avoidance,

and traffic anomalies (e.g., weather conditions).

- ▶ *N2N*: this layer focuses on the navigation information inside a zone, considering the graph representation. The expected features consist of zone graph management, node navigation data sharing, path planning processing, and emergencies decision-making, for instance, an emergency pathway for a drone that broadcasted an SOS message due to propellers failure;
- ▶ *E2E*: Although the N2N layer handles the navigation management inside a zone, the E2E focuses on the inter-zones management. Hence, it is expected that this layer implements the routing of drones between adjacent zones managing the shared access to the gates, the hand-off of adjacent zones, and inter-zone traffic congestion notifications.
- ▶ *Service*: While the three above-mentioned layers focus on management and fair access to the airspace, the Service layer aims to provide a shared platform where drones can broadcast their service requirements. For instance, a given drone can notify other drones that it will perform a service of some corporation in a specific region. As the message will be encapsulated, only the drones that pertain to the same corporation will understand the message, through the next presented layer.
- ▶ *Application*: As the name suggests, this layer focuses on the applications provided by drones. From the ZSP point of view, the Application layer is a black box whose content is decapsulated only by the drones involved in the related application.

As presented in Figure 2.2b, protection mechanisms must ensure the required security and privacy levels throughout all the layers. A major IoD challenge is that security is mostly provided for the application layer rather than the lower ones [5]. Furthermore, drones can demand different levels of privacy, depending on the provided service [7]. For instance, a drone delivery service may allow the drone's location at a certain level with the service customers. On the other hand, a surveillance service must require full cloaking regarding the drone's position. Thus, the service-oriented level of security and privacy is an open challenge in IoD.

Table 2.1: Comparison between traditional mobile networks and the IoD

Aspect	IoD	Cellular	Vehicular	WSN	UWSN
Speed	High, typically over 15mps	Slow, less than 5 mps	High (on highways), over 20 mps; medium (on urban ways), around 8mps	Slow, typically following the integrated device speed	Slow, following the ocean/river dynamics
Mobility	Varies according to the application	Random	Follows the road layout	Typically random	Follows the ocean/river dynamics
Altitude change	High	Small	Small	Small	Medium
Topology change	High	High	High	Low	Medium
Comm.	A2A, A2G, commonly at the LoS, outdoor	G2G, environmental obstacles, indoor and outdoor	G2G, environmental obstacles, outdoor	G2G, environmental obstacles, indoor and outdoor	underwater acoustic signals, commonly at the LoS
Processing power	Medium	High	High	Low	Low
Energy constraints	High	Medium	Low	High	High

2.3 The Relation Between IoD and Traditional Mobile Networks

IoD is categorized as a mobile network, likewise Vehicular, Cellular, Wireless Sensor (WSN), and Underwater Sensor (UWSN) Networks. Table 2.1 presents a comparison of different aspects between these networks. IoD has particular characteristics that differ from the other networks, representing a novel mobile network paradigm. IoD is unique in terms of mobility since the drones fly at a high speed over the airspace, being able to change their altitude several times during the performed service. Also, given a source and a destination point, a drone's path planning can be changed with less impact compared to other networks, such as VANETs. In this case, the ground vehicles must follow the road layout. On the other hand, the IoD network infrastructure can define new airways at a minimum cost, allowing drones to perform pathway deviations [19].

Likewise in Cellular and VANETs, the IoD topology changes fast. Also, drones can communicate with other drones through RF or optical-based Air-to-Air (A2A) channels as well as with terrestrial nodes through Air-to-Ground (A2G). One of the paramount IoD communication advantages is the LoS propagation, providing a faster transmission with less packet losses [1]. However, as well as the nodes of UWSN, drones are spread over different altitudes, representing a challenging environment in terms of communication protocols. Furthermore, the communication channel is more susceptible to attacks compared to the other networks. Although the LoS provides a series of advantages, it allows the performance of continuing eavesdropping, which can lead to other harmful attacks, such as jamming, spoofing, and hijacking [9].

Moreover, drones have energy constraints, which limit their processing power and flight time [14]. IoD is unique regarding this aspect. As the Cellular and Vehicular network nodes, drones move fast gathering and exchanging data through embedded devices (e.g., cameras, microcontrollers, and a plethora of sensors). However, their battery power is much more constrained than cars, for instance. Allied with these issues, the drone's propellers consume a high amount of power, differing the consumption model from a node of WSN or UWSN [20].

2.4 Concepts of Security and Privacy in the IoD

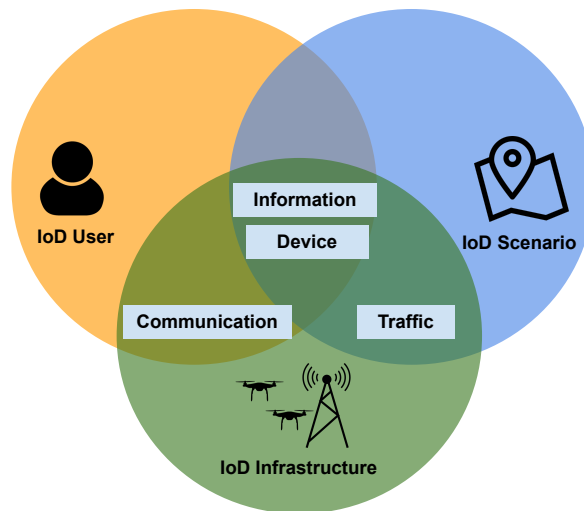
Security is a fundamental aspect of computer networks, entailing a set of strategies and protocols designed to protect digital assets against an extensive range of potential threats. It encompasses several requirements, such as confidentiality, integrity, and availability, ensuring the preservation of information, and permitting access solely to duly authorized entities [9]. Privacy is a paramount security aspect that must also be ensured in any network. It pertains to the protection of users' sensitive information, communication patterns, and location data as they traverse mobile networks [7]. With the ubiquity of mobile devices, such as smartphones and sensor devices, maintaining privacy is crucial to preventing unauthorized access, eavesdropping, and data breaches.

Considering the traditional concept of privacy applied to human users, there are four categories that encompass personal privacy [21]: information, bodily, communication, and territorial privacy. The first category is related to the personal data that can be obtained by an attacker; the second concerns the safety of the human body against not allowed procedures; the third involves the level of privacy in all types of human communication and; the fourth covers the limits of intrusion on personal space.

These categories can be easily fitted to a UAV instead of a human. However, it is necessary to consider that, in information and territorial privacy, human (represented by the user) privacy merges with the UAV since drones carry the user's personal information with them. Based on these categories, we define IoD privacy and its four major drone-centered privacy issues, presented as follows.

- **IoD privacy:** It is defined as the right of any IoD component – for instance, a drone, a ZSP, or a user – to decide how, when, and for which purposes their information could be released to external parties;
- **Information-based privacy:** It refers to the safety of the drone's data, for instance, its identity, location, and sensitive information related to the provided ser-

Figure 2.3: Venn diagram of IoD privacy issues and their relation with the IoD components



Source: Elaborated by the author

vice;

- **Device-based privacy:** This category points out the precaution regarding the IoD devices, roughly, considering the hardware. It also involves the methods that a drone needs to handle in case of failures or atypical conditions;
- **Communication-based privacy:** It concerns the information leakage over the communication protocols between the network nodes;
- **Traffic-based privacy:** It pertains to the civilian laws and traffic regulations that must be respected in different scenarios. This category also involves people's and places' privacy, which can be violated by the drone, e.g., a flight over a forbidden region;

The privacy issues are strictly interlaced with three main aspects of the IoD environment, which are: the user, whose personal information and intentions are carried by the UAV nodes and can be stored in a cloud service; the scenario, where the drones are flying and, consequently, can be damaged or violated by external elements; and the infrastructure, responsible for ensuring the security of information that is related to the network.

Figure 2.3 illustrates a Venn Diagram of these relationships. Personal information leakage directly affects the user and/or a UAV node. It can occur through the scenario – for instance, physical eavesdropping of a human attacker – or through the infrastructure, e.g., a Spoofing Attack (SA). Likewise, device-based privacy is related to the three aspects. The scenario may present factors that affect the drone's performance. Exemplifying, adverse weather conditions can damage the UAVs, causing a network rupture. In addition,

Table 2.2: IoD application aspects and what privacy issues they impact

Application aspect	Privacy Issue			
	<i>Information</i>	<i>Communication</i>	<i>Device</i>	<i>Traffic</i>
Airspace			●	●
Grounded environment			●	●
Localization	●			●
Communication channel	●	●		
Other drones	●	●		
Network infrastructure	●	●	●	
Third-party network	●	●		
Users	●	●		
Sensors and camera	●	●	●	●
Operational software	●	●	●	

drones are susceptible to “environment attacks”, such as a hijacking attack in which an attacker can take down a drone and obtain its data (including user’s information).

Communication-based privacy involves the security mechanisms to keep the exchanged data between user vs. infrastructure and infrastructure vs. infrastructure safe. Aside from data safety, these mechanisms must consider some aspects of IoD. For instance, a UAV is, generally, an embedded system with energy and fuel constraints. In this context, the study and application of lightweight communication protocols are mandatory.

Traffic-based privacy is the most changeable category: the civilian and traffic regulations may vary from city, state, or country. Also, the infrastructure must know these variations, handling its topology to fit the scenario demands. For instance, consider two scenarios: (a) a big city with several high-rise buildings; and (b) a small city in the countryside. In (a) the drone’s altitude is probably the main privacy concern since the people living in the buildings can be “surveilled” by the drones. On the other hand, this concern is a minor issue in the (b) scenario.

Privacy issues are also interlaced with IoD applications. Each aspect represents a target for an ill-intentioned entity that aims to harm the network, as presented in Table 2.2. Likewise discussed in Figure 2.3, information is a crucial privacy issue, having the potential to be explored through the majority of aspects, ranging from a simple localization spot to detected failures over the operational software, that can forfeit a task.

Communication is also a compromised issue by a bulk of application aspects. Each party that demands a data exchange is a potential target to a malicious attacker through its communication channel. Device and traffic privacy issues have a smaller number of aspects that can lead an attacker to affect them. However, information and communication issues are more critical to the application. Airspace and the grounded environment impose different conditions on a drone during a service – e.g., a thunderstorm – directly affecting its hardware integrity and route.

2.5 Chapter Remarks

This chapter presented the fundamental concepts of the Internet of Drones (IoD). We formally defined the main components of this mobile environment, and compared the main aspects of IoD with the traditional mobile networks, discussing the similarities and differences between them. This comparison reinforced the assumption that IoD is a new environment with particular characteristics.

Moreover, this chapter discussed the main concepts regarding security and privacy, linking these concepts with IoD. As presented, the IoD particular characteristics pose a new scenario from a security/privacy point-of-view, opening a vast research field to investigate. Due to these issues, several protocols and mechanisms applied to traditional mobile networks may not be adequate to IoD. For instance, protection mechanisms ensure security and privacy based mainly on both the environmental characteristics and the node's configuration [11].

In the next chapter, we present the main attacks involved with this paradigm, the current protection mechanisms to avoid these attacks, and why they potentially can not embrace the IoD characteristics properly, reinforcing the research questions related to this dissertation.

Chapter 3

Attacks and Protection Mechanisms in IoD: Overview and Design Guidelines

This chapter discusses the attacks that threaten the IoD environment and the existent protection mechanisms that can potentially mitigate these attacks. From this discussion, we propose a framework to guide the design of novel IoD PMs.

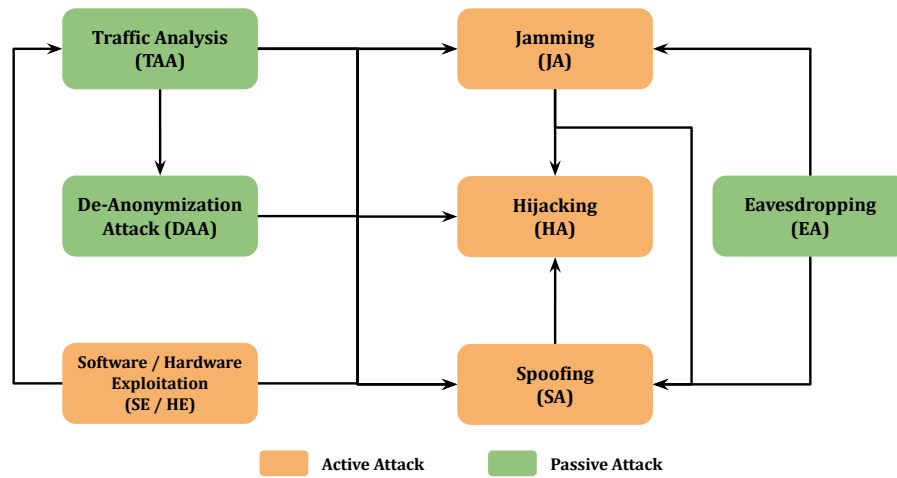
The chapter is organized as follows. Section 3.1 surveys seven major classes of attacks that affect IoD. In Section 3.2, we list six groups of potential PMs that can mitigate the surveyed attacks, and how they can protect the IoD elements. We point out the main trends and challenges regarding the design of PMs for IoD in Section 3.3. From this discussion, we propose a framework to guide the Design of IoD-based PMs in Section 3.4. Section 3.5 presents our chapter remarks.

3.1 Attacks in IoD

In the previous chapter, we discussed security and privacy aspects related to IoD, considering four drone-centered issues. This new scenario leads us to analyze and classify UAV-related attacks from a new perspective, surveying the attacks that can deeply affect the network due to its particular characteristics. It is important to state that, in this Dissertation, we focus on investigating the attacks that can be favored by these particular characteristics, highlighting the new challenges imposed by the attacks. Exemplifying, a replay attack is a type of threat that can occur in UAV networks, however, the considered IoD model does not cause any changes in how this attack can occur, or even the damage level caused in the network [4]. On the other hand, Jamming Attack has several differences in IoD, as will be discussed in Section 3.1.4.

We surveyed seven major classes of attacks that deeply affect IoD: Traffic Analy-

Figure 3.1: IoD-related attacks taxonomy



Source: Elaborated by the author

sis Attack (TAA); Software/Hardware Exploitation (SE/HE); De-Anonymization Attack (DAA); Jamming Attack (JA); Eavesdropping Attack (EA); Spoofing Attack (SA); and Hijacking Attack (HA). These attacks can be grouped according to the presented behavior, as passive or active.

► **Definition #1: Attack behavior** Indicates if an attack has the potential to break some level of privacy and apply damages to IoD by itself (active attack) or if it just contributes to improving the knowledge about the IoD environment, acting as a passive attack to an active attack.

Mobile networks, in a general way, can be affected by a variety of attacks, from different sources and purposes. Due to its characteristics, IoD is more susceptible to some attacks when compared to other networks. Likewise, some attacks do not present considerable damage. In the same way, a range of them can be considered as a privacy-based attack – reaching the sensitive data as the primary goal – and another portion has the potential to affect privacy, but it is not their goal.

UAV-centered attacks are a serious threat in military environments, leading to the possibility of government-sensitive data leakage and, in critical situations, a diplomatic crisis. In 2011, a military drone from the United States was captured by the Iranian forces in Kashmir city. According to an Iranian engineer interview¹, the drone was hijacked through a JA followed by a GPS Spoofing Attack (GPS-SA), forcing the drone to land on Iranian’s ground instead of to land on the US military base.

Some attacks are commonly developed using as prior knowledge the results from another privacy attack. Another important aspect is that they commonly do not damage the drone device or the infrastructure. In many cases, they are carried out in a “silent

¹<https://www.csmonitor.com/World/Middle-East/2011/1215/Exclusive-Iran-hijacked-US-drone-says-Iranian-engineer>

Table 3.1: Main aspects of the IoD-based attacks

Attack	Type			Mobility		Privacy Issue			
	<i>Single</i>	<i>Multiple</i>	<i>Smart</i>	<i>Stationary</i>	<i>Mobile</i>	<i>Information</i>	<i>Comm.</i>	<i>Device</i>	<i>Traffic</i>
DAA			●	●		●			●
TAA			●	●	●	●			●
EA	●	●	●	●	●	●	●		
JA	●	●	●	●	●		●		●
SA	●	●	●	●	●	●	●	●	
HA		●	●	●	●	●	●	●	
SE/HE	●		●	●		●	●		

mode”, intercepting the sensitive data without the network noticing the attack. Furthermore, an active attack can act as an intermediary to another, improving the effectiveness of the whole attack. These relations are indicated by means of arrows in Figure 3.1. Thus, a “workflow” of attacks can be designed. For instance, a TAA can provide network knowledge serving enhanced information to SA. This attack, in turn, can be used as the first door to other attacks, such as HA.

Improving our taxonomy, Table 3.1 presents fundamental characteristics regarding the attacks. The type of an attack α indicates what are the most common ways that α is performed: if it is performed by itself, in a single way (single); if a variation of α is applied at the same time in order to improve the effectiveness of the attack (multiple); or if different attacks are performed together, with different computational methods, having α as the main attack (smart). Mobility refers to the most common mobile behavior of the malicious entities related to α . Finally, we also summarize what privacy issues are affected by α .

In the next sections, we discuss in detail the seven named attacks. Besides the topics of Table 3.1, for all of them, we introduced a formal definition, its relation with other mobile networks, its current literature overview, and the challenges involved.

3.1.1 De-Anonymization Attack (DAA)

De-anonymization is a technique that breaks the data’s anonymization regarding a given entity, such as people and cars. De-anonymization is strictly related to the discovery of identity, also called re-identification. However, the technique does not embrace the identity only but the interlaced characteristics of each entity. For instance, anonymous grounded vehicles can have their identity discovered by analyzing their mobility behavior [22].

► **Definition #2: De-Anonymization Attack (DAA)** Let us consider a set of attackers $A_{DAA} \subset \mathcal{A}$, a set of IoD nodes \mathcal{N} , a set of IoD anonymized entities \mathcal{N}_{anon} , a set of cyber tools \mathcal{CT} , a bijective function f that maps $n \rightarrow n'$ such $n \in \mathcal{N}$ and $n' \in \mathcal{N}'$, and a bijective function f' that maps $n' \rightarrow n$. DAA occurs when A_{DAA} applies \mathcal{CT} to get the function f' in a way that f' is the inverse function of f . In other words, A_{DAA} tries to reach the correct relation between the anonymized and the actual data.

The performance of a previous attack is mandatory to perform DAA, generally, the one that leaks the IoD data at some level. Exemplifying, a SE can be explored to reveal stored data. Based on this data, a TAA has the potential to aid an attacker in executing DAA. In turn, DAA is also applied as an intermediary attack. For instance, it can lead to discovering a specific drone or user, so the attacker intends to direct their final attack. Therefore, DAA represents a threat to both information and traffic privacy issues.

DAA has been extensively studied in traditional mobile networks, mainly in Vehicular and Cellular networks. The dissemination of social networks and the LBSN, such as Facebook, Foursquare, and Waze, also contributes to the improvement of the DAA model, in which public user information can be combined with their mobility, typifying a Location Privacy Leakage (LPL). LPL embraces information and traffic privacy as a “multi-privacy issue” [23].

In the LPL concept, there are a bunch of DAA-based attacks, called Location Privacy Attack (LPA), whose goal is to de-anonymize the network entities through attacks over their location. Besides the user and/or vehicle identity leakage, these attacks entailed the re-identification of sensitive places (called Points of Interest (PoI)), trajectories, and transportation modes behavior [22, 23] of targeted users.

To our knowledge, there is no study modeling DAA in IoD scenario. It is essential to clarify that we are considering the attacks that directly affect the IoD nodes/users, excluding studies that only use data from drone-based services, e.g., re-identification of grounded vehicles provided by video surveillance of traffic roads [24]. The central aspect that contributes to this absence is the lack of anonymized drone mobility trace datasets, either real or synthetic. Hence, developing these datasets represents an excellent research opportunity, allowing a series of related studies.

As previously discussed, drones do not transport people and, consequently, they do not lead to IoD user’s mobility. On the other hand, it can have a great relationship with IoD users. Exemplifying, drone-based delivery services carry people facilities, commonly having as the final destination the user location. Thus, supposing that a DAA is successfully performed over a set of drones, is it implied in de-anonymizing the IoD users? All these subjects are unexplored areas in IoD, creating a significant challenge to solve.

3.1.2 Traffic Analysis Attack (TAA)

Traffic Analysis Attack (TAA) is a passive attack whose primary goal is to extract enhanced knowledge about a network based on data gathered by the node's mobility. Due to its particular characteristics, this approach is applied in mobile networks with a high level of nodes with a well-defined mobility pattern, for instance, Vehicular Networks. As well as ADD, considering the general context of mobile networks, TAA is not applied as an attack commonly, being used to improve several aspects of a mobile network and its environment as a whole [25].

► **Definition #3: Traffic Analysis Attack (TAA)** Let us consider a set of drones \mathcal{D} , a set of attackers $A_{TAA} \subset \mathcal{A}$, a set of cyber tools \mathcal{CT} , a time interval Δ_t , a network knowledge level kl , and a function $f(kl, \mathcal{D}, \mathcal{CT}, \Delta_t)$ that calculates the knowledge gain regarding IoD to A_{TAA} after Δ_t . TAA occurs when $f > 0$.

In the literature, TAA is an under-investigated IoD-related attack. As presented in Section ??, Sciancalepore et al. [26] integrated traffic analysis as a primary method for the automated detection and identification of drones. Several collected network data, such as the number of packets and the inter-arrival time, are used to classify and predict different aspects regarding the network nodes' traffic.

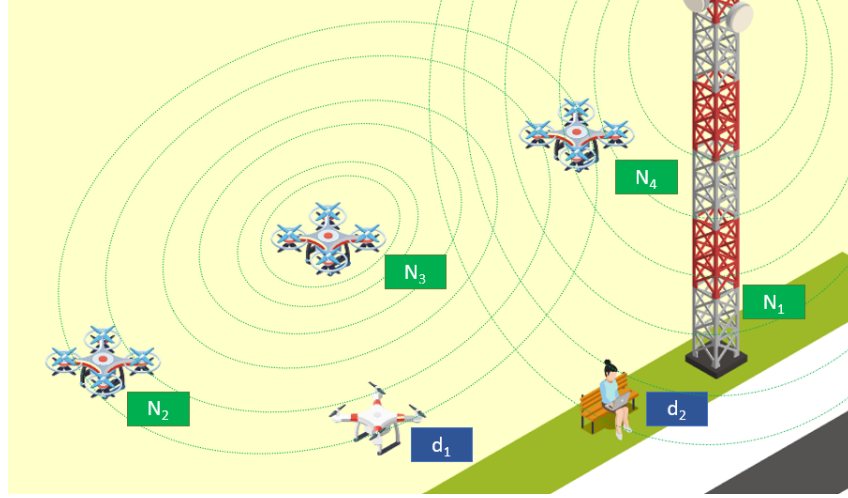
The lack of well-established airways in practice is a fundamental aspect that contributes to the absence of studies that embrace TAA. Traffic analysis becomes a challenging task without traces of real-world drones, and consequently, TAA can not be studied thoroughly.

3.1.3 Eavesdropping Attack (EA)

The cyber Eavesdropping Attack (EA) exploits the communication link between two or more network nodes, violating the network reliability. In an unauthorized way, a malicious node infiltrates a network through some communication vulnerability and monitors the data exchange. In the same way as JA, an eavesdropping attack can also be applied as a protection mechanism, discovering and avoiding malicious users over a network [27, 28].

► **Definition #4: Eavesdropping Attack (EA)** Let us consider a set of attackers $\mathcal{A}_{EA} \subset \mathcal{A}$, a set of IoD nodes \mathcal{N} , a communication link λ , a network knowledge level

Figure 3.2: An example of EA being performed



Source: Elaborated by the author

kl , and a function $f(\mathcal{A}'_{EA}, \mathcal{N}', \lambda, kl)$, such $\mathcal{A}'_{EA} \subset \mathcal{A}_{EA}$ and $\mathcal{N}' \subset \mathcal{N}$, that calculates the knowledge gain of the network during the communication. The EA occurs when $f > 0$, which means, a communication among nodes was successfully intercepted, improving the attacker's knowledge.

Figure 3.2 illustrates an example of EA carried out by two malicious entities – a drone d_1 and a grounded attacker d_2 – over a set of network devices, the base station n_1 and the drones n_2 , n_3 , and n_4 . The green circles represent the communication radius range of transmitters n_2 (on the left) and n_1 (on the right). We can note that d_1 is inside the communication area of n_2 and, thus, can perform an EA. Similarly, d_2 is inside the n_1 communication range. We assume that both d_1 and d_2 get unauthorized access to the IoD network.

Acting directly over network communication, EA represents a severe threat to information privacy, in which sensitive data can be intercepted. Eavesdropping is a well-explored approach in traditional networks.

Following this, many proposals considered that the EA model presented an evolution regarding its target based on the communication channel. Another important evident aspect observed is the single attack modeling. The studies of Xiao et al. [29] and Khan et al. [30] are the only ones that perform an attack and have as target a Drone-to-Everything (D2X) communication. This latter exploited untrustworthy parties where personal data can be eavesdropped on aiming to perform active attacks, such as hijacking.

Among those studies, five of them have as a target a D2I communication. Liu et al. [31], Liu et al. [32], Zhang et al. [33], and Fan et al. [34] modeled the attack occurring over a communication channel (the first and second studies considered a wiretap channel) in which the attacker is a stationary grounded node. Similarly, Ma et al. [35] presented a model with stationary grounded eavesdroppers. However, the scenario consists of a UAV-

enabled mmWave relaying network in which a link is disconnected, and the communication occurs through a selected relay.

The communication between drones (D2D) as a target was explored in the preliminary study of Hoang et al. [36], where a single communication between two actual network drones has eavesdropped through an unauthorized drone. With a more robust model, Liu et al. [37] considered a drone swarm scenario. In this model, drones as eavesdroppers try to exploit the communication of several drones (acting as transceivers). In contrast, a swarm of drones acts as intermediate drone relays, assisting the communication among the transceivers.

EA's research also investigated Drone-to-Ground (D2G) communication in which the grounded nodes are not part of the infrastructure. Cui et al. [38] considered an attack model where a drone communicates with mobile grounded nodes susceptible to eavesdroppers on the ground. The application of drones as a base station of wireless networks (UAV-BS) is a current research trend. Kang et al. [39], Lei et al. [40], Cheng et al. [41], and Wu et al. [42] modeled an attack scenario in which wireless users act as eavesdroppers in random positions under a UAV-BS coverage area. Also, Kang et al. [39] considered the presence of buildings and solid obstacles in the coverage area, simulating an urban scenario.

Included in the eavesdropping context, there is a specific variation of attack called Man-in-the-middle Attack (MiMA), whose goal is to intercept a legitimate communication between two nodes, disrupting the actual connection and acting as an intermediary point. MiMA can actively modify the sensitive data transmitted without the nodes' awareness. MiMA represents a severe threat to IoD regarding information and communication issues. However, only some studies consider this attack with a defined model, as in the attack types above [43]. In both studies, the attacker is assumed to be powerful enough to read, modify, and replay the intercepted messages. The central aspect that can lead to this lack of knowledge is that the IoD topology is very dynamic, with drones moving in different altitudes at a high speed, which can make the performance of MiMA. Notwithstanding, MiMA is an attack that deserves special attention and further investigation since it can significantly affect IoD privacy.

3.1.4 Jamming Attack (JA)

Jamming Attack (JA) is a type of Denial of Service (DoS) that occurs when a set of devices neutralizes communication between network nodes by flooding the network communication channels. The main goal is to ensure that any node cannot use the network,

hampering the provided service. JA is a threat to IoD privacy regarding communication and traffic issues since it can hamstring the communication channel and, consequently, influence mobility and the drone's traffic [44].

► **Definition #5: Jamming Attack (JA)** Let us consider a set of attackers $\mathcal{A}_{JA} \subset \mathcal{A}$, a set of drones \mathcal{D} , a threshold τ , a communication link λ , and a function $f(A'_{JA}, \mathcal{D}', \lambda)$, such that $A'_{JA} \subset \mathcal{A}_{JA}$ and $\mathcal{D}' \subset \mathcal{D}$. The function f calculates the flooding caused by A'_{JA} in λ . JA occurs when $f \geq \tau$, indicating that there is no possibility to the IoD nodes to communicate.

From a device point of view, a drone can be used as a jammer device to hamper the communication of different mobile networks. Likewise, a drone-based network can be exploited by a jamming device. In the literature, JA has been investigated in the IoD context. Recently, some studies have explored this area, improving the state-of-the-art.

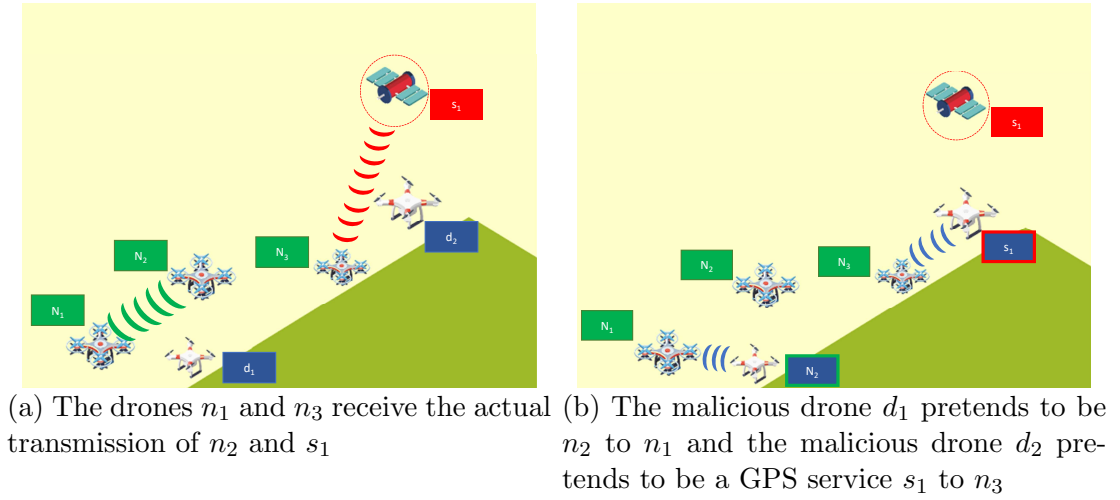
Some studies investigated JA in a broad scope: Xu et al. [45] modeled a JA with a smart jammer, adjusting its strategy adaptively; Li et al. [46] discussed two varieties of jamming attacks based on the speed of the malicious drone. On the other hand, some studies focused on specific UAV scenarios. Xiao et al. [29] modeled JA, focusing on D2I communication. Van den Bergh and Polin [47], as well as Gupta et al. [48], described the drone's susceptibility regarding a Global Navigation Satellite System (GNSS) jamming. Sliti et al. [49] investigated JA considering Optical communication.

Several studies investigate how an attacker affects the drone's mobility [44, 50, 51, 52]. Overall, they discussed the difficulties faced by a jammed grounded base to planning the trajectory of drones first, explaining how a malicious drone can nullify the benefits of a network with a multi-path routing protocol, and showing how D2I communication can hamper the drone's mobility and its power consumption.

Regarding the attack type, most are focused on performing a JA only, demonstrating punctual investigations that consider the specific characteristics of the attack. On the other hand, inspired by the advent of Software Defined Network (SDN), there are proposals [29, 45, 46, 50] that exhibited an innovative approach in which the malicious entity has an attack pool and different parameters that are considered to perform a proper JA over the network. Various aspects are considered in the smart choice method, such as the environment communication power [29, 50], the attacker's ability to infer the location and time of a drone's transmission [45], and the drone's speed [46].

Mobile jammers are investigated in most proposals to model a realistic scenario, trying to nullify the network communication at different levels. The studies that state a stationary jammer only justify this mobility pattern by considering that their main goal is to hamper the network, centralizing the attack on the ZSP component whose mobility is also stationary. An important aspect is that only two studies have considered stationary and mobile jammers [46, 47].

Figure 3.3: An example of SA being performed



Source: Elaborated by the author

3.1.5 Spoofing Attack (SA)

Spoofing Attack (SA) is a kind of network falsification where a malicious attacker pretends to be an authorized network node or an authorized network service. Figure 3.3 illustrates an example of SA, where two malicious drones, d_1 and d_2 , falsify their identities as authorized nodes/services in the network. Drone d_1 pretends to be n_2 when communicating with n_1 and tries to obtain sensitive data regarding the network. Meanwhile, the drone d_2 performs a GPS-SA, acting to provide the GPS service originally provided by s_1 . Thus, d_2 can change the planned path of n_3 .

► **Definition #6: Spoofing Attack (SA)** Let us consider a set of attackers $\mathcal{A}_{SA} \subset \mathcal{A}$, a set of drones \mathcal{D} , and a set of services \mathcal{DS} that can be provided by \mathcal{D} or a third-party network service. The SA occurs when an attacker $\alpha \in \mathcal{A}_{SA}$ falsifies its identity i_α in which a node $d \in \mathcal{D}$ believes that $\alpha \in \mathcal{D}$ or $\alpha \in \mathcal{DS}$.

SA is a critical threat to IoD privacy, in which sensitive information can be leaked or changed through communication disruption. The most exploited network target is the GPS signal. The GPS-SA is one of the most imminent threats against civilian drones since most GPS services have no protection mechanisms used on the transmission signal [53]. A malicious entity can perform other attacks through this attack, such as hijacking drones, transmitting fake GPS signals, and forcing a drone to land on a suspicious field.

Shepard et al. [53] presented the first comprehensive study of GPS-SA, discussing that this attack is effective over the civil GPS signal when the attack has previous knowledge regarding the GPS signal. In this study, the authors discussed different devices that can perform the attack and the attack architecture, including various strategies based on

the drone's distance. Until then, there was no defense mechanism against GPS-SA, a high-priority study to be studied in the security context. The SA model established in that study was widely adopted in subsequent proposals investigating SA and protection mechanisms to avoid it [54, 55, 56, 57, 58, 59].

The attack type and mobility of the SA model are similar in most GPS-SA studies, which demonstrates once again the influence of Shepard et al. [53]. The attacker entity is represented as a stationary GPS spoofer, performing a single GPS-SA attack over drones inside the coverage attack area. The study of Zhang et al. [58] is the only one that models multiple attacks with a mobile spoofer, considering a malicious drone that performs eavesdropping and a GPS-SA. Another prominent aspect is that the attack is executed over a single drone. Huang and Wang [56] and Eldosouky et al. [60] improve this aspect by attacking simultaneously in a group of drones.

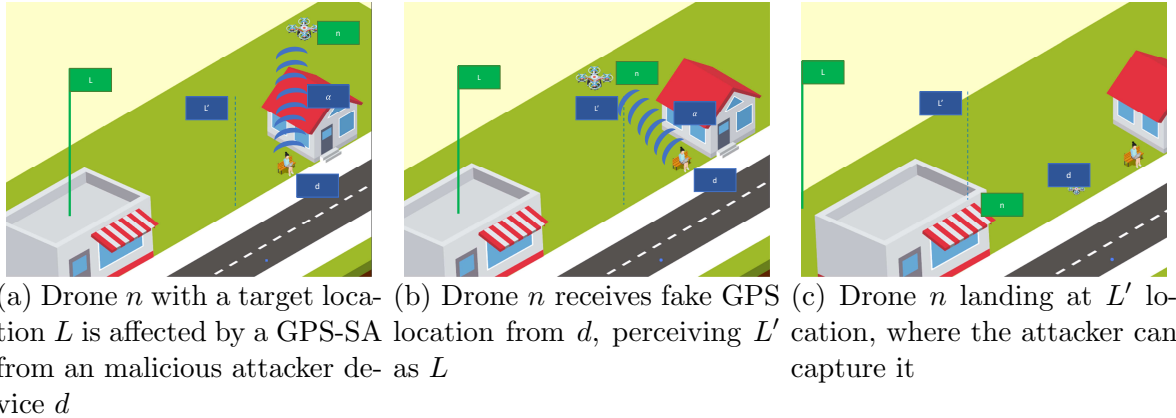
Two relevant studies exploit different targets. As presented in Section 3.1.4, Xiao et al. [29] modeled an attack that can perform an SA based on previous knowledge obtained in an eavesdropping attack. In this model, the spoofer communicates with a targeted drone, masquerading its identity to obtain sensitive information about the network and/or the specific drone. Davidson et al. [61] exploited the downward-facing optical flow camera used by some drones to stabilize the flight. To perform this attack, an attacker needs to cover and spoof three main aspects: the environmental influence, in which the attacker must be able to alter the appearance of the ground plane that the optical flow camera captures; the plausible input, referring to the attacker's influence over the environment to induce a sensor's reading used as valid input; and the meaningful response, meaning the control that the adversary has over the device.

3.1.6 Hijacking Attack (HA)

Being one of the most dangerous threats to IoD, the Hijacking Attack (HA) tries to control a specific drone, forcing it to move to an attacker's geographic location of interest. A primary attack is commonly necessary to accomplish the hijack, for instance, a JA, a SA, or an EA (or all).

► **Definition #7: Hijacking Attack (HA)** Let us consider a set of attackers $\mathcal{A}_{HA} \subset \mathcal{A}$, a set of drones \mathcal{D} , a set of preliminary attacks Ω , a geolocation $dest$ that represents the drone's final destination, and a function $f(\Omega, \mathcal{D}', dest)$, $\mathcal{D}' \subset \mathcal{D}$, that indicates a new calculated region $dest'$ to drones \mathcal{D}' . The HA occurs when $f \neq dest$, in other words, there is a change of the actual drone's destination, indicating an interception.

Figure 3.4: An example of HA being performed



Source: Elaborated by the author

Figure 3.4 illustrates an example of HA in which an attacker intercepts a drone, pretending a localization L' instead of the actual localization L . In this situation, a GPS-SA is accomplished as an intermediary attack to manipulate the drone's location.

HA demands a great effort from the attacker, requiring a series of favorable factors in a specific time interval to accomplish it. Considering military tasks performed by drones, this kind of attack has profound intentions, in which the attacker tries to hijack the drone to obtain governmental data or even destroy him. In civilian contexts, HA still demands a great effort to be accomplished. However, the attacker's intentions vary according to the application context and target.

Over the years, different studies have investigated HA in the civilian context. Currently, HA is performed following an approach in which the attack model considers different factors, chooses a set of intermediary attacks to take control of the drone, and hijacks the device. The other two prominent aspects pointed out in the table are the stationary behavior of an attacker and the GPS signal as a primary target of exploitation. Unlike others, Mendes et al. [62] is the only one that discussed the possibility of performing an HA using a support malicious drone in addition to the grounded attacker.

As discussed in Section 3.1.5, most GPS services have no protection mechanisms used on the transmission signal, which leads to a concentration of attacks. Several studies [55, 62, 63, 64, 65] showed how the GPS-SA leads to a HA

Besides the GPS-spoofing, both the communication channel and malicious software are well-explored means to perform HA. Mototolea and Stolk [66] investigated the DSM2 radio protocol aiming to achieve an attack over small drones. Daubert et al. [67] explored Telnet and MAVLink protocols with the same objective. Some studies investigated the failures and vulnerabilities of specific devices. Pleban et al. [68] described an HA performed over the AR.Drone 2.0 by exploiting security vulnerabilities, such as port scan and backdoor software, and combining different cyber attacks. Kang and Joe [69] explored the Erle-Copter drone through communication link exploitation. Similarly, Jares

and Valasek [70] described an HA through the communication feedback signal. The adversary performs the attack by changing the feedback signal to send a false state to the victim. Thus, the drone goes to the attacker’s target state rather than the original one.

Analyzing the current scenario of IoD, its applications, and the literature, there is a lack regarding the HA as a threat to delivery services. Besides the risk of the drone’s hijacking, the packages are also an attacker’s target. This factor leads to a novel approach to theft, in which an attacker steals the goods carried by the drones.

3.1.7 Software and Hardware Exploitation (SE/HE)

Software / Hardware Exploitation (SE/HE) mainly aims to exploit network data, affecting both information and device privacy issues. They are the “first door” to cyber-attacks, exploring previous network infrastructure failures. This aspect makes SE/HE a complex threat to counter since its traceability is harder to detect and has a broad attack range. For instance, if a drone’s operational software is released with a backdoor, all IoD applications that have this drone as a device are affected by an SE. Likewise, if a drone’s radio transceiver presents a factory defect, it can be explored to break the network’s privacy.

► **Definition #8: Software Exploitation (SE)** Let us consider a set of malicious attackers $\mathcal{A}_{SE} \subset \mathcal{A}$, a set of IoD nodes \mathcal{N} , a set of drone-related softwares \mathcal{DS} , such that $\forall n \in \mathcal{N}$ \mathcal{DS} is part of n , a set of cyber tools \mathcal{CT} , and a function $f(\mathcal{CT}, \mathcal{DS}, \mathcal{N})$ that calculates the level of information privacy leakage regarding \mathcal{N} . SE occurs when $f > 0$, which means, \mathcal{A}_{SE} develops or changes a software $s \in \mathcal{DS}$ in a way that the network information is leaked in some level.

► **Definition #9: Hardware Exploitation (HE)** HE can be defined similarly: let us consider a set of malicious attackers $\mathcal{A}_{HE} \subset \mathcal{A}$, a set of IoD nodes \mathcal{N} , a set of drone-related hardware components \mathcal{DH} , such that $\forall n \in \mathcal{N}$ \mathcal{DH} is part of \mathcal{N} , a set of cyber tools \mathcal{CT} , and a function $f(\mathcal{CT}, \mathcal{DH}, \mathcal{N})$ that calculates the level of device privacy damage regarding \mathcal{N} . The SE occurs when $f > 0$, which means, \mathcal{A}_{HE} develops or changes some behavior of a hardware $h \in \mathcal{DH}$ in a way that it is damaged, affecting the network.

The literature presents some reports regarding using SE as an intermediary attack of HA. Kang and Joe [69] described the exploitation of backdoors and port-scan of AR.Drone 2.0 and Erle-Copter, respectively. The oneCase Cheerson CX-10W, DJI Phan-

tom 3, and Hawkeye II 2nd FPV Quadcopter are also devices with software failures [71]. If these devices have some hardware failure, it can be explored through the previous SE.

SE is not limited to drone software. Malicious software stemming from a third-party client can also intercept sensitive data from an attacker or even cause damage to the network. Considering that the IoD is strictly related to the IoT and has data storage in the cloud, if a SE is performed successfully on the cloud side, an attacker can obtain all services' vital information, including the user preferences and the mobility traces of the drones.

Desnitsky and Kotenko [72] investigated the application of both SE and HE, aiming to hamper the drone device through energy depletion attacks, whose focus is to exploit its autonomy through the wasting of the communication channel and/or its motor movements. They discussed the different targets to perform this attack, which include the primary drone's movements, namely, takeoff, landing, and free flight; as well as the obstruction or overload of communication channels through intermediary attacks, such as Jamming Attack (JA) and Spoofing Attack (SA) (described in sections 3.1.4 and 3.1.5, respectively).

SE and HE are even a serious threat to other mobile networks. Hasrouny et al. [73] pointed out that VANETs are susceptible to SE and HE, where data integrity and authenticity can be violated, mainly through malware injection over the vehicle's main system. Ferrag et al. [74] highlighted that some cellular network protocols are vulnerable, allowing privacy data leakage. The same aspect occurs in WSNs and, considering that there are e-health WSNs, SE and HE represent a high-priority threat to be handled.

3.2 Protection Mechanisms for IoD: Do We Need to Reinvent the Wheel?

Every network paradigm must take into account its associated security requirements to become a safe and robust environment and protect its infrastructure and users. There are fundamental concepts that define and specify all these requirements, such as availability, integrity, and confidentiality [9]. In the previous section, we presented which attacks threaten the IoD environment and which elements they affect, potentially. Thus, in this section, we focus on the existent PMs to mitigate these attacks.

Firstly, we present in detail which elements demand protection, grouping them into four different categories and highlighting their relations. As previously discussed, some attacks can occur not in IoD only, but also in other mobile networks, for instance, the

JA and SA. Likewise, there is a range of protection mechanisms to avoid these attacks in different network paradigms. Thus, we highlight one of the fundamental questions of this dissertation: can these existent protection mechanisms provide an adequate protection level in IoD? Do they fit in this novel environment properly?

3.2.1 Elements to Protect in IoD

Considering the recent advancements of IoD [1, 4, 9], this network paradigm has several elements that demand protection. They can be summarized into four categories: device, communication, environment, and information. These elements have several relationships, which can influence the required level of security and how the protection mechanisms must be designed.

Table 3.2 presents the elements that demand protection and their relationships, denoted through the black dots. Each row has an element, which is associated with a label. Each black dot arranged in its row indicates a relationship with a previously defined element, labeled by the top of the column. In this way, it is possible to find relationships between previously described elements and the next ones. For instance, the *Battery* element (*D2*) is related to the *Sensors* element (*D1*), defined earlier. Looking at the column with its label, *D2*, it is possible to find the other relationships with the other elements of the *Device* and the *Communication* categories, besides the *Trajectory*, *Data Storage*, and *Data Processing* elements.

Table 3.2: IoD elements that demand protection and their relationships

Category	Demands protection
Device	Sensors (<i>D1</i>)
	Battery (<i>D2</i>)
	Propellers (<i>D3</i>)
	Radio (<i>D4</i>)
	Drone Apps. (<i>D5</i>)
	ZSP Apps. (<i>D6</i>)
Traffic	POIs (<i>E1</i>)
	Trajectory (<i>E2</i>)
	Weather (<i>E3</i>)
	Animals (<i>E4</i>)
	Vehicles (<i>E5</i>)
	Buildings / Stations (<i>E6</i>)
	People (<i>E7</i>)
Comm.	Channel Config. (<i>C1</i>)
	MAC (<i>C2</i>)
	Routing (<i>C3</i>)
	Encryption (<i>C4</i>)
	Authentication (<i>C5</i>)
Info.	Drone Location (<i>I1</i>)
	3rd-party Location (<i>I2</i>)
	Data Privacy (<i>I3</i>)
	Data Storage (<i>I4</i>)
	Data processing (<i>I5</i>)

► *Device*: this category embraces the protection of both hardware and software of the entire network. It is necessary to ensure that the embedded devices (from drones to other nodes, such as base stations) work according to the manufacturer's specifications and what action must be taken in case of malfunction. Likewise, the drone application services must meet the invasion-proof criteria, ensuring a proper security level for the drone. Moreover, it is expected that the drone has applications exclusively dedicated to communicating with the ZSP, demanding a high-priority security level. The battery has relationships with all the other elements, highlighting that any battery's failure or even an abnormal power consumption can malfunction the other devices.

► *Traffic*: here, we highlight the elements beyond drones that can be affected by the drone's traffic, such as aerial animals, weather conditions, and those representing network nodes: industrial vehicles, buildings/stations, and people. Likewise, the drone must protect these elements, avoiding harming them. For instance, if a given drone presents a malfunction when flying, how to prevent it from falling over a person? The PoIs and the trajectories are central elements influencing the drone's trajectory. Therefore, they also affect the other mobile nodes since they communicate with the drone.

► *Communication*: it is associated with the communication channel requirements that demand a higher security level, ranging from the physical to the upper layer. It includes the proper configuration for transmitting and receiving messages, ensuring that the transceivers will process them accordingly. Furthermore, it is necessary to encrypt the communication channel and authenticate the involved parts as best as possible, avoiding malicious entities taking advantage of the exchanged messages. These elements are totally tied with the battery device since the SWaP limitations demand the development of lightweight protocols. Furthermore, communication security is related to all environmental mobile nodes. If the communication channel is violated, so are the mobile nodes of the network.

► *Information*: it is a valuable item that demands protection. Besides the communication based policies, information must follow privacy guidelines. For a given drone's service, it is necessary to define what level of location and other sensitive information can be shared from both drone and third-party users, which affect their security. Furthermore, the IoD system needs to have well-defined approaches to storing and processing the data, ensuring a proper security level for all network components.

In this section, we summarize six different categories of PMs: LPPMs, ADD, anti-jamming, anti-spoofing, cryptographic-based techniques, and air traffic regulations. ADD and Air traffic regulations focus on UAV-based networks. Anti-Jamming, Anti-Spoofing and cryptographic-based mechanisms provide protection for different mobile

Table 3.3: Protection mechanisms and what element groups they protect

PM / Elem. Group	Device	Traffic	Communication	Information
LPPM	●	●		●
ADD	●	●		
Anti-Jamming		●	●	
Anti-Spoofing		●	●	●
Cryptographic-based	●		●	●
Air traffic regulations		●		

networks, including UAV-based ones. However, LPPMs have not been designed for IoD yet, representing an open challenge.

Together, these categories can cover the element groups that demand protection, as presented in Table 3.3. Furthermore, these mechanisms can mitigate the discussed attacks, as shown in Table 3.4, where a half-filled circle indicates partial protection against the attack, and a full-filled circle indicates complete protection against a given attack. We discuss each PM in the next sections, extending the discussion regarding the aforementioned tables.

3.2.2 Location Privacy Protection Mechanisms

Location privacy is a fundamental concern in mobile networks. Since a mobile node commonly shares its location to obtain information from an LBS, it is vulnerable to being leaked. It can cause severe damage to the network, exposing sensitive data of them, such as their identity, and spatial and temporal location. Location privacy is a well-studied area in mobile networks, such as VANETs. Different privacy models have been developed to ensure data confidentiality. On the other hand, several threats related to localization can break the protection mechanisms, such as TAA.

Due to IoD being an envisioned scenario, there is a lack of studies regarding location privacy in this environment [1, 7]. However, location information is omnipresent in IoD. Considering the prospection that this environment will spread over urban and rural scenarios, it is mandatory to investigate the current Location Privacy Protection Mechanisms (LPPMs) and whether they meet the IoD characteristics. Although there are no LPPMs designed for IoD, specifically, we discuss four well-studied categories related to this type of mechanism, referring to the other mobile networks.

► **Anonymization-based LPPMs:** Several mechanisms are based on the node's identity change using pseudonyms. They are based on the *k-anonymity* concept, whose

Table 3.4: Protection mechanisms and what attack categories they can mitigate

PM / Attack	TAA	DAA	SE/HE	EA	JA	HA	SA
LPPM	●	●			●	●	●
ADD				●	●	●	●
Anti-Jamming					●	●	●
Anti-Spoofing				●	●	●	●
Cryptographic-based		●	●	●	●	●	●
Air traffic regulations				●			

idea is to avoid an attacker to identify individuals from a small subset of their attributes, diminishing the probability of re-identification and, therefore, their location [11].

A classical anonymization-based LPPM is Mix-Zones [75], whose goal is to change the pseudonyms when k nodes enter and leave a geographic region r at a close time interval. When a node is inside the Mix-Zone, it cannot communicate with nodes outside the zone. From an attacker’s point of view, Mix-Zones cause “confusion” in the track of both node’s identity and location since the trajectories before and after the zone are “unlinked”. They are well investigated in traditional networks, mainly VANETs [75, 76, 77, 78, 79].

Two questions regarding the design of Mix Zone-based mechanisms are: *Where are the best regions to place a Mix Zone?*; and *How many Mix Zones are necessary to provide proper location privacy?* They represent the problem known as the Mix Zone Placement (MZP) [80]. Typically, the current mechanisms place Mix Zones in regions where a considerable concentration of nodes throughout a specific time interval. In VANETs, for instance, they are placed near road intersections controlled by traffic lights [76]. However, these places do not exist in an IoD environment. Furthermore, the concentration of drones can vary depending on different factors, such as the drone’s task and the density of the drone’s traffic.

► **Spatio-temporal obfuscation:** This strategy also considers the *k-anonymity* concept to “increase” the granularity of spatial information instead of the exact location. Thus, a node reports its approximated location area. If two or more nodes are near each other, they can report the same place, creating cloaking regions, and obfuscating their actual position. Therefore, an attacker can not obtain the location of nodes with high accuracy. There is a range of obfuscation-based mechanisms designed for traditional networks [11], and most of them apart from a seminal study that introduced the *l*-diversity concept [81]. This approach considers at least l different geolocations to create the *k-anonymity* group. Thus, is harder for an attacker to assume where each anonymized user is since the locations are distant enough from each other. Nonetheless, the l selected geolocations must be semantically coherent, allowing the mechanism to be able to provide an adequate level of privacy.

- ▶ **Dummy-based mechanisms:** These mechanisms aim to generate a group of dummy queries simulating fake nodes so that third-party servers, e.g., a malicious LBS, cannot distinguish the real node. Also, they are well investigated in traditional networks, but there is no study regarding these mechanisms in IoD [82, 83, 84, 85]. They are strictly related to environmental factors, mainly the communication model, and both the node's trajectory and traffic flow. As discussed, IoD differs from the other networks regarding these characteristics.

- ▶ **Protocol-based mechanisms:** Differing from the above categories, a protocol-based mechanism aims to provide location privacy without modifying the node's data [86]. Its premise is to apply a node-to-node privacy protocol, based at most on the nodes' proximity. LPPMs included in this category have the best privacy performance in *ad hoc* mobile networks. Considering the IoD, protocol-based mechanisms require an accurate study. They are commonly based on cryptographic protocols that demand a significant power of processing while drones have SWaP constraints.

3.2.3 Anti-Jamming Mechanisms

Over the years, JA has been countered in different wireless and mobile networks through anti-jamming mechanisms [87]. Likewise, some studies assess these mechanisms for UAV-related networks [44, 51, 88]. These works analyzed how JA interferes in the drone's trajectory, discussing how UAVs communicate with grounded nodes and can suffer a jamming signal from a grounded, stationary source. QoS represents a key aspect of designing anti-jamming mechanisms, in which communication throughput and delay are main requirements considered.

Commonly, anti-jamming strategies model the jamming threat as an optimization problem, designing a trajectory planning method to optimize the drone's position in the airspace dynamically, introducing new concepts (e.g., slack variables allied to two-block coordinate descent (BCD) algorithms [51]) to solve it sub-optimally, which improved significantly different QoS requirement levels. Drones are also considered as a relay communication node in other mobile networks (such as VANETs) in the model of anti-JA [89]. Recently, Machine Learning-based strategies have been considered for intelligent anti-jamming mechanisms for UAV-based networks [88, 90], being based on reinforcement learning and federated learning approaches.

None of these approaches considered an environment with a robust and interoperable network where the drones fly following well-defined airways, as designed in the

IoD concept. The constrained flyable airspace designated by the airways is a prior requirement that was not considered in the previous works. All studies assume that the drones can fly freely over the airspace and, consequently, regions free from the action of the JA can be reached. However, drones have limited airspace to fly in IoD. Given that an airway is affected by a JA in which it does not have a flyable region free from the attack, the drone's path planning must be reformulated. Considering the current solutions, for instance, the trajectory optimization proposed by Wu et al. [51], the drones trajectory will deviate aiming to optimize its transmit power and the minimum delay considering the original path planning. However, this deviation can lead the drone to invade other airways that were not originally in its flight plan. These aspects pose severe risks to the availability and integrity of other drones since collisions can occur, representing open challenges to be solved.

Furthermore, it is expected that a given airway has a constant traffic flow in a real-world environment. Inductively, the identification of a compromised airway leads to a path planning reformulation of all drones that will fly over this airway. The system must handle these issues and communicate with the affected drones as soon as possible. These aspects pose serious risks to IoD, mainly regarding the drone's availability, representing open challenges to investigate.

3.2.4 Automatic Drone Detection

Nowadays, government entities have been expending efforts to regulate airspace (e.g. the Remote ID rule [91]). However, the lack of a standardized flight policy facilitates the presence of unauthorized drones. Although these unauthorized drones may just be flying with no intention to cause damage, they can be malicious entities, performing passive attacks, for instance, eavesdropping communication, or causing serious damage, such as UAV hijacking.

In addition to human vision-based drone recognition, there is a wide range of phenomenologies to detect a drone as well as the technologies that use them. As phenomenologies, we have the reflectance of photons and radar, acoustic and electromagnetic emission, and the induced magnetic field. The technologies consist of passive visible and thermal imaging, active time of flight systems, acoustic-based sensors, Radio Frequency (RF) and radar-based systems, magnetic detection systems, and human intelligence. Over the years, the drone automatic identification problem has been improved with the exploration of smart approaches integrated with traditional technologies and methods, such as the use of Deep Learning algorithms [26, 92, 93, 94].

It is important to state that ADD is composed of two different tasks: (i) detection, in which an environmental element is detected as a drone; and (ii) identification, referring to what kind of drone is the detected one. The identification can occur in a general way, where the proposed mechanism just infers if the element is a drone and, moreover, if it is authorized to fly in that space. On the other hand, identification can be comprehensive, informing the drone’s brand, number of motors, distance, and others.

ADD acts in the “front” of protection, trying to identify an anomaly in the IoD (in this case, an unauthorized element). Hence, this approach provides partial protection facing a range of attacks, as described in Table 3.4. Nonetheless, other actions must be taken when ADD detects some unauthorized element. These methods can be very useful to avoid aerial animals since an ML-based model can be trained to differentiate drones from aerial animals [95]. Furthermore, ADD is a key mechanism to avoid early attacks performed by unauthorized drones with a significant impact in IoD, such as EA and SA.

Identification is not a task handled by the majority of the studies. It means that the proposed techniques aim to detect the presence/absence of a drone in the environment, but do not concern with accurately identifying what a drone is. However, in a real-world IoD environment, there is a prospect that several models of drones from different companies will fly cooperatively. Hence, the correct identification of drones will be mandatory to avoid properly malicious ones, which demands the investigation of robust protection mechanisms.

3.2.5 Anti-Spoofing Mechanisms

SA targets the GPS signals, mostly. Considering that drones will fly without a human controller in IoD, these signals are vital to the drone’s flight over the airways, requiring accurate protection. The current anti-spoofing mechanisms are mainly based on signal processing techniques. Autonomous integrity monitoring techniques have been applied to ensure safer navigation to UAVs [96, 97, 98]. They assess the availability performance of the GPS signals by calculating the protection level on-the-fly, which is the radius of a circular area centered around the position solution [97].

Furthermore, ML-based techniques can mitigate significantly the occurrence of SA in UAV-based networks [59, 99, 100]. These studies consider the environmental aspects where the drones are involved in terms of the GPS signal, for instance, jimmer, shimmer, and frequency modulation. Apart from the detected environmental factors, the best ML-based method from a pool of models, such as K-learning, and SVM.

3.2.6 Cryptographic-based Mechanisms

Currently, cryptographic-based mechanisms embrace a wide range of methods to ensure authentication, mainly, providing a high level of security. Commonly, these mechanisms protect the communication channel. Differing from LPPMs, several studies designed drone-centered cryptographic protocols. Most of them consist of key agreement schemes, involving both the drones and authority entities [30, 101, 102].

Considering the range of trending technologies in recent years, blockchain is undoubtedly one of the most successful cryptographic-based approaches, being applied in several fields. Blockchain consists of a distributed system based on the mutual trust between the parties applying different concepts, such as hashing, smart contracts, consensus protocols, and public and private keys [103]. In the UAV-based networks, blockchain can ensure security and privacy in several ways, also in some aspects already discussed: mitigating jamming signals; detecting possible hijacking situations; avoiding collisions; authenticating the involved nodes; and protecting data dissemination [103].

Compared to traditional network environments, most state-of-the-art authentication techniques do not fit appropriately in IoD requirements since drones have SWaP limitations. Cryptographic-based mechanisms demand a significant computational effort that can compromise the drone's energy efficiency. Hence, IoD-based authentication mechanisms must take into account this issue, verifying previously if the mechanism represents a lightweight approach in terms of computational processing, affecting as less as possible the drone's autonomy to perform a given service. Recently, novel approaches have been proposed to overcome these issues. A technique that has gained attention is the Physical Unclonable Functions (PUF) [104]. According to its hardware features, it represents a unique and "unclonable" physical identity of a device. PUF can provide reliable authentication for D2X communications.

3.2.7 Air Traffic Regulations

As grounded vehicles that follow transit laws, drones must move under air traffic regulations in an ITS environment. Since drone usage is an emerging area, the rules regarding a robust drone's air traffic are in the initial steps. A reference entity for air traffic is the Federal Aviation Administration (FAA) in the United States. Nowadays, the FAA has proposed some advances toward uniform unmanned aircraft system traffic management. For instance, FAA announced the implementation of the Remote ID, a

framework followed by an in-flight drone to provide identifying information that other parties can receive [91]. Technically, Remote ID consists of a standard message that drones have to broadcast from their takeoff to the shutdown. The following elements compose the standard Remote ID message:

- ▶ **Unmanned Aircraft (UA) ID:** it refers to a unique ID for the drone, being its serial number or a session ID;
- ▶ **UA and Control Station Location:** it is composed of the latitude, longitude, altitude, and velocity for both the drone and the control station;
- ▶ **Emergency Status:** when a drone has some anomaly, or it is in a risk situation, it flags an emergency in this data;
- ▶ **Time Mark:** this data is given by a synchronized timestamp of the system;

The message's broadcast must occur periodically, with a maximum time interval of 1 second. Also, the guidelines define the Industrial, Scientific, and Medical (ISM) frequencies band to be used in the available communication channel. Regarding the Remote ID deployment, there are two ways to meet the requirements. Firstly, the drone's manufacturer can build a drone with a built-in standard remote ID capability that follows the final rule's requirements. Otherwise, an individual Remote ID's broadcast module must be attached to the drone. Thus, the company side that is operating the drone can be able to see the vehicle at all times during the flight. Furthermore, a drone can operate without the Remote ID broadcast when it is flying over FAA-recognized identification areas, commonly defined by community organizations or educational institutions.

In Europe, new civil aviation regulations have been proposed and applied by the EU Aviation Safety Agency (EASA), conceiving a regulatory framework for secure drone operations. However, the proposed framework is centralized in a few authorities, and their rules are mostly unclear. Hence, there is a need to review the framework, delegating decentralized tasks [1].

The lack of well-established legislation represents uncertainty for the IoD development at all levels. Traffic regulations contribute to avoiding some attacks, mainly the ones related to the drone's flight. On the other hand, they directly impact the drones' mobility, thus, influencing location privacy. Another challenge is the possibility of having different traffic regulations in different places. In this case, a location privacy approach needs to be as general as possible, specializing its implementation according to each regulation.

3.3 Trends and Challenges

As discussed, there are a set of attacks that have their models based on their similarities to other mobile networks, for instance, JA, SA, and EA. Therefore, it is intuitive that the existing protection mechanisms can properly counter them. However, the IoD particular characteristics make room for new exploitation targets. The continuous development and definitions regarding IoD infrastructure as well as the airways regulation have the potential to change IoD environment constantly, posing a need to study, review, and explore both the attacks and protection mechanisms. In the following, we list the major trends related to the occurrence of new attacks in IoD, and the challenges in the design of IoD-centered protection mechanisms.

► **Drone’s Mobility Traces and Location Privacy Attacks.** Node’s mobility trace is fundamental to characterizing and analyzing the mobility behavior of a given network environment. This behavior is a useful way to perform a DAA over VANETs [22]. However, IoD scenario (in contrast to VANET), in practice, still is not well established, hindering the experimentation of related studies. Hence, there is an absence of drone mobility traces available. This absence represents a big challenge since it affects the investigation of different mobility-dependent fields, e.g., location-based privacy attacks as well as location-based protection mechanisms. Still considering a comparison with VANETs, there are several attack and protection models well defined and investigated in grounded networks that are not studied in IoD, mainly because there are no mobility traces available as an information source to these approaches. A suitable way to address a drone’s mobility traces is through synthetic traces, generated by IoD simulators.

► **Energy Constraints.** This aspect is a fundamental issue related to drones, imposing a great challenge not solely in the security/privacy context but also in communications, path planning, and other fields. This constraint affects significantly how an attack/protection mechanism is modeled. Robust attack models tend to process a massive amount of data to obtain a high level of success, and, consequently, consume more energy. The same behavior occurs with smart and collaborative protection mechanisms. In the former case, research directions must concentrate on design strategies to delegate the potential drone’s data processing aiming for a data balance, or even sharing the processing with other network nodes, with greater energy efficiency. This challenge leads to the following one.

► **Decentralization of AI-based Approaches:** the advancement of AI-based techniques also embraced the design of protection mechanisms, providing accurate detection

of potential adversaries [92]. However, most of them demand a high computation effort to properly train and deploy in a real-world scenario. It faces the drone's SWaP limitations, which can bring an impracticable scenario. Nonetheless, AI-based mechanisms can be decentralized through the network nodes with a higher computational power, such as those belonging to stationary devices, and the Cloud System. Furthermore, the UAV-assisted Federated Learning [105] is a promising approach that can be explored toward this decentralization.

► **Smart Models, Mobility and Environment Prediction.** A major risk in IoD environment is the continuous improvement of smart attacks. In HA discussion, for instance, we can observe that intermediary attacks can lead to a better performance of hijacking in which different environmental factors need to be analyzed together. In addition, there are considerable network aspects with the potential to affect the attack performance (including the “intermediary attacks choices”), e.g., drone's energy constraints, being another challenge pointed out. Therefore, there is room for the development of smart attacks and, consequently, important challenges to be considered in the design of a protection mechanism. For instance, the attacker's mobility also represents an opportunity in terms of attack modeling. Thus, considering smart attacks, how to decide the mobility behavior? Does a stationary attacker perform a required level of success? On the other hand, if a mobile attacker is modeled, what is its impact in terms of the system constraints?

In a nutshell, an intelligent protection mechanism must be composed of different countermeasures. The intelligent observation of environmental factors can serve as the initial step in implementing the mechanisms as well as the mobility behavior of IoD nodes. Thus, they must be activated dynamically based on the prediction of an imminent attack situation. Furthermore, the decentralization of these mechanisms is a key requirement since IoD embraces a wide and heterogeneous environment, where sensing in different locations can deeply contribute to deploying mitigation techniques.

3.4 A Framework to Guide the Design of IoD-based Protection Mechanisms

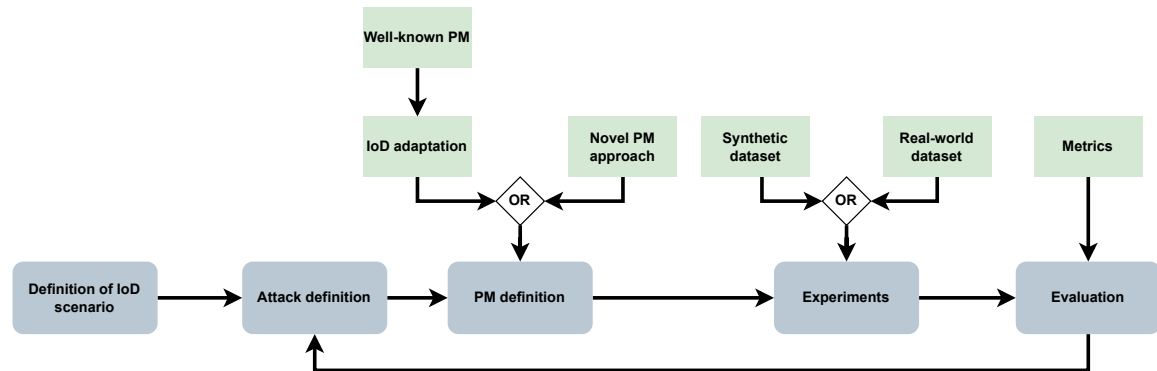
The discussion of this section highlighted a major aspect: unprecedented attacks can occur in IoD, or even well-known attacks can occur in a different manner. This aspect entails in several challenges for the design of protection mechanisms in which a major question is to investigate if the existing mechanisms can provide the same protection

level to IoD compared to the traditional mobile network environments. The two research questions (RQ1 and RQ2), presented in Section 1.1, are totally tied to these challenges.

Therefore, we propose a framework to guide the design of protection mechanisms for IoD, representing the key directions toward security and privacy. This contribution handles the current challenges in this area since it is still in its first steps. It is necessary to explore novel defense mechanisms and evaluate the protection mechanisms' performance stemming from other mobile networks. Figure 3.5 shows our proposed framework considering a flow chart layout, ranging from the first step to the last one. In other words, the figure presents the flow, step-by-step, that a researcher may follow to design an IoD-based protection mechanism. They are described in detail as follows.

- ▶ **Definition of IoD scenario:** the first step consists of defining the specific scenario of interest, including mostly the environment definition (e.g., urban, rural, industry environment), and the service/application scope (e.g., on-demand delivery or surveillance). Furthermore, the required QoS levels shall be defined in this step (besides the security/privacy requirements). For instance: is the application delay-tolerant? what is the minimum acceptable level of the drone's energy to perform the service?
- ▶ **Attack definition:** based on the scenario definition, the attack design must be in accordance with the drone's services, defining the goals and which elements can be exploited. Formally, this step consists of mathematically defining the threat model. Exemplifying, let us suppose the primary goal of the designed defense mechanism is to provide location privacy through anonymization. In that case, the attack needs to propose a cybernetic approach whose objective is to re-identify the network nodes and/or users.
- ▶ **Protection mechanism definition:** next, it is necessary to design the protection mechanism based on the following approaches: by proposing a novel approach; or by investigating an existing one. The design must consider the current literature allied with the two former steps. The first case is the best strategy to follow when there is a lack of knowledge regarding both the investigated scenario and the attack model. On the other hand, the latter shall be applied when there is a need to study the impact of the IoD characteristics in the observed scenario and attack, for instance, JA or SA. Two or more approaches can be combined to perform better security/privacy levels.
- ▶ **Experiments:** after defining the attack approach and the protection mechanism, we need to plan the experiments comprehensively. A critical issue involves the dataset and the simulation tools to carry out the experiments, which is a challenge in this area. Also, it is mandatory to follow a well-structured experimentation methodology, ensuring its correctness and replicability.

Figure 3.5: Framework to guide the design of IoD protection mechanisms



Source: Elaborated by the author

- **Evaluation:** finally, it is necessary to define the metrics to evaluate the proposed mechanism. Understanding the evaluation results can lead to a new cycle of experiments. A series of adjustments in the scenario, the mechanism, and the attack might be necessary until the results point out that the mechanism's performance is suitable to the proposed IoD scenario.

3.5 Chapter Remarks

This chapter presented the main concepts regarding seven major IoD-related attacks. For each attack, we formalized it through a mathematical notation, establishing a general threat model and contributing to a concise definition of them. Given these attacks, we discussed what elements are necessary to protect in IoD and what are the existent mechanisms to protect them and mitigate the attacks, surveying six different categories of mechanisms.

Bearing all these aspects in mind, we discussed that most of these mechanisms do not meet the IoD particular characteristics, presenting a potential lack of security and privacy in this environment. Therefore, we proposed a framework to guide the design of IoD-centered protection mechanisms, embracing the following steps: the definition of the IoD scenario; the attack and protection mechanism definition; the experiments; and the evaluation. This framework meets the scientific method but inserts specialized aspects into each design step.

In the further chapters, we design novel protection mechanisms for IoD following the proposed framework. They are grouped into three categories: LPPMs, anti-jamming, and ADD. For each proposed mechanism we: raise the application scenario; define a

threat model of the attack; assess if there is an existent mechanism to mitigate the attack and what adaptations must be made; carry out thorough experiments; and evaluate the solutions using well-known security and privacy metrics.

Chapter 4

Design of Location Privacy Protection Mechanisms for IoD

This chapter presents the contributions related to the design of Location Privacy Protection Mechanisms, representing the major research front of this dissertation. We design three new LPPMs: t-MixDrones, MixRide, and TDG. The design of these mechanisms is guided by the proposed framework in such a way that all of them can overcome the performance of the existing mechanisms, being applied in different situations. Considering that they provide a suitable level of location privacy in different environmental conditions, we model an RL approach for their dynamic assignment, providing enhanced levels of location privacy regardless of the IoD network conditions. Furthermore, we present a case study regarding the importance of location privacy in the IoD context, and how the existing governmental policies must be in accordance with these principles.

This chapter is organized as follows. Section 4.1 brings an introduction and motivation regarding the importance of designing new LPPMs for IoD. Sections 4.2 and 4.3 present the design of two Mix Zone-based LPPMs for dense environments: t-MixDrones and MixRide, respectively. In Section 4.4 we present the design of TDG, a mechanism proposed for sparse environments. Section 4.5 discusses the fundamentals of IoDAPM, an RL-based approach for the dynamic network assignment of the proposed mechanisms. In Section 4.6, we technically demonstrate through the design of a location-based attack that the Remote ID rule, proposed by the FAA, is a threat to the drone's location privacy considering its final proposal. To tackle this issue, we propose an enhanced design of Remote ID, incorporating different privacy-related mechanisms in its model. Lastly, Section 4.7 brings the chapter remarks.

4.1 Introduction

DAAAs represent a critical issue in the IoD research field since they are a major group of location privacy attacks (discussed in Section 3.3). Considering that this class of attacks has not been investigated in IoD, one of the major research fronts of this dissertation is to study the attack model of DAA, the existing mechanisms, verify if they are suitable to IoD, and therefore, propose new mechanisms. Specifically, we investigate comprehensively the trajectory-centered attacks, whose goal is to de-anonymize the drones' identity through traffic analysis, characterizing a TAA. Hence, the proposed mechanisms face a smart attack in the IoD environment in which TAA act as a primary attack to DAA.

As discussed, there is a lack of studies regarding the use of this kind of LPPM in the IoD. Most of the proposed strategies take advantage of terrestrial vehicular behavior to provide the required privacy level. For instance, MZ strategies consider road intersections controlled by traffic lights as a proper place to consider as the MZ region [76]. However, aerial traffic does not have this characteristic. These aspects reinforce the need to study the current strategies in this network paradigm, assessing the need to adapt them to IoD.

We design three new LPPMs considering different IoD scenarios. For dense environments, we propose two MZ-based mechanisms: t-MixDrones and MixRide. While t-MixDrones focuses on providing a suitable privacy level through maneuvers for altitude changes, MixRide aims to improve the energy efficiency of drones through collaborative rides between aerial and ground transportation systems, overcoming the shortcomings highlighted by t-MixDrones. As IoD can be deployed in sparse environments, we design TDG, a dummy-based mechanism that can provide a suitable level of anonymization regardless of the density of drones in the airspace.

Although all of these mechanisms present suitable levels of protection in terms of location privacy, they were designed for specific scenarios. Therefore, we can assume that no mechanism can be considered a "silver bullet" for addressing optimal location privacy. Therefore, we model an RL approach for the dynamic assignment of the proposed mechanisms in a given IoD environment to cover the design of smart mechanisms, highlighted as one of the major challenges in IoD security/privacy research field.

The next sections present the design of each LPPM in detail, following the framework proposed in Section 3.4.

4.2 t-MixDrones

The t-MixDrones is a traffic-aware MZ-based LPPM for urban scenarios with a high density of drones, taking advantage of their’ “omnipresent” behavior. As they move fast and can support the communication of different mobile nodes, it is possible to expand the applicability and integration with other networks, providing new IoT solutions, and embracing the new era of communication technologies, strongly supporting connectivity through 5G and beyond (B5G).

The proposal of t-MixDrones considers the prospected scenario of different drone-related companies performing heterogeneous services that demand a constant information exchange with 3rd-party users [1]. For instance, drone delivery must keep the final users informed about the remained flight time until the delivery as well as an approximated location. Therefore, we assume that the sensitive information of drones must be anonymized to preserve the integrity of both drones and users.

Considering this scenario as a starting point, we follow the remained steps of the guidelines framework to design t-MixDrones. Firstly, we formulate the threat model. After, we present the basic concepts of MZ and the recent related studies that proposed MZ-based PMs in existing mobile networks. From that, we formally design t-MixDrones. With the previous steps completed, we define a methodology to perform experiments and, therefore, carry out an extensive evaluation through simulations. Each step is described in the next subsections.

4.2.1 Threat Model

The main goal of a set of adversaries \mathcal{A} is to de-anonymize the successive pseudonym changes of a given drone. The mobile node trajectory is a potential aspect to be exploited by a malicious entity[9, 79, 105]. In this case, \mathcal{A} can perform trajectory-based linking attacks. In this type of attack, \mathcal{A} tries to link the generated sub-trajectories, rebuilding the full path planning of a targeted drone. Once \mathcal{A} links each sub-trajectory correctly, it can also link the pseudonyms to a single drone.

An inductive question arises about performing trajectory-based linking attacks: How to link the sub-trajectories correctly? As discussed, drones have energy constraints that affect their mobility and flight time. Hence, we formulate the hypothesis that, given a geographic destination $v \in G.V$ (from the general airspace graph model), the drone will fly following the shortest path over the airways $G.E$. This is a reasonable assumption

considering the SWaP limitations of drones, mainly the power restrictions.

In this study, we assume that a given attack performed by \mathcal{A} has the following steps [22]:

1. The attack calculates the minimum path using the Dijkstra algorithm for all combinations of trajectories before and after the Mix Zones application;
2. It calculates the Dynamic Time Warping (DTW) between the minimum path and the trajectory combination;
3. It minimizes the attribution costs generated by DTW considering a cost matrix. The attributions designate the matching between a previous and subsequent pseudonym associated with trajectories;

Furthermore, we make the following assumptions about adversary \mathcal{A} [1, 9, 79, 105]:

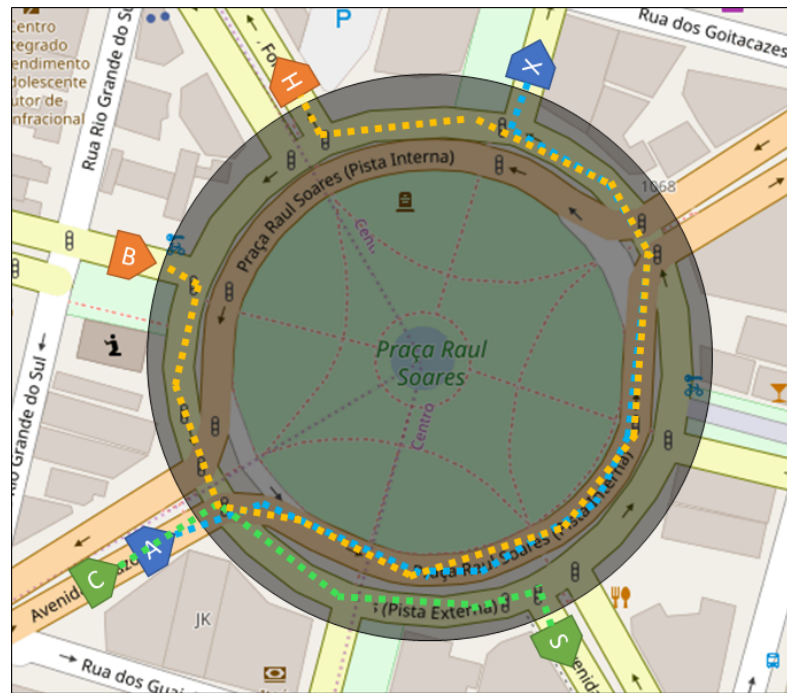
- \mathcal{A} has a background knowledge kl , addressed by the exploitation of a location server $\mathcal{LS} \in \mathcal{MN}_{IoD}$ that manages and stores the drones requests;
- \mathcal{A} has complete access to \mathcal{LS} as an eavesdropper, but not as a *man-in-the-middle*. It implies that although \mathcal{A} can access all location requests and information sharing of the drones, it can not modify them;
- \mathcal{A} has complete airway topology knowledge. In other words, \mathcal{A} can build a topological graph $G = (V, E)$ of the airways;
- The IoD network protection mechanisms are not aware of \mathcal{A} 's existence. It means that there is no action to break its silent performance;
- The communication channel of a given ZSP $z \in \mathcal{Z}$ and the authorized drone $d \in \mathcal{D}$ is fully reliable and cannot be violated, ensuring the protection of the exchanged data.

Given these assumptions, t-MixDrones must be designed to mitigate the effect of \mathcal{A} over the IoD network \mathcal{MN}_{IoD} , mainly over the drones as mobile nodes.

4.2.2 Mix Zones Concepts

MZ is an approach to provide location privacy for mobile entities in a mobility environment through pseudonym changing [106]. A pseudonym is an identity that acts as a protector mechanism, hiding the entity's real data. MZ can be informally defined

Figure 4.1: Example of a MZ region



Source: Svaigen et al. [80]

as a geographic region that changes pseudonyms (pseudonymization) of k entities that are inside it. Thus, users enter the MZ with a pseudonym, change to a new unused pseudonym, and, after a length of time, exit under the new pseudonym [106]. Since pseudonymization occurs with k entities, it is intuitive that MZ must be placed in regions with heavy traffic of entities.

Figure 4.1 illustrates the MZ method applied to ground vehicles as mobile entities. In this example, a MZ was placed at a roundabout, and the shaded circular region indicates the coverage area. Three different cars are near the MZ with pseudonyms “A, B and C”. Let us assume they entered the MZ at a close time. In that way, a pseudonymization was processed, giving them new pseudonyms: “X, H and S”, respectively. If an attacker eavesdrops on this service, obtaining the user’s trace data, when “A, B and C” enter the MZ region, the attacker can not obtain inside-data. Thus, the next obtained data will have the new pseudonyms, causing difficulty to the attacker’s trace monitoring. This will probably hamper the attacker’s success.

Besides the location privacy protection level, the MZP is a current challenge linked with the design of MZ-based mechanisms, consisting of modeling how many MZs are necessary to perform the anonymizations and where to place them [80]. They should be placed in strategic regions that maximize the anonymization coverage in an optimal configuration. Theoretically, it can be addressed by covering all the possible regions. However, the MZ management demands a high computational cost, being an unfeasible solution. Thus, this task lays on an NP-Hard problem [107].

Table 4.1: Recent studies of MZ mechanism

Ref.	Network	MZP Approach	Explored Features
[112]	VANET	Reputation-based pseudonymization	– Road intersection
[107]	ITS	Multiple MZ for Map Services	– Vehicle’s position
[113]	VANET	Cluster-based pseudonym change	– Vehicle’s energy – Vehicle’s position
[80]	VANET	Clustering and Exponential Moving Weighted Average	– Vehicle’s position
[114]	ITS	De-correlation privacy model	– Traffic analysis – Traffic light – Parking location
[115]	ITS	Dynamic swap zones	– Vehicle’s position
[116]	ITS	ML-based vehicle’s lane changing	– Traffic analysis
[117]	VANET	Decoy traffic	– Road intersection – Vehicle’s direction
[118]	VANET	Indistinguishability Swap	– Vehicle’s speed – Vehicle’s position
[119]	ITS	Privacy preserving authentication	– Communication delay
[120]	ITS	Differential privacy	– Traffic analysis – Parking location
[76]	VANET	Real-time traffic-based MZ	– Traffic analysis – Traffic lights
[77]	VANET	Neural Network-based attack model	– Traffic analysis
[75]	VANET	Communication delay-based MZ	– Roadside infrastructure
Ours	IoD	– Bio-inspired MZP – Airways as a mixing factor	– Traffic Analysis – Drone’s position – Airways congestion

4.2.3 Related Studies

Over the years, several studies investigated MZ in mobile networks, introducing and defining solid guidelines for advancing the state of the art regarding location privacy [108, 109, 110, 111]. Recently, MZ has been massively explored in VANETs and ITSs as a whole. These studies treated both traffic and roadside features to place and apply the MZs. IoD, in turn, is almost an unexplored environment regarding the use of MZ as a protection mechanism. In fact, t-MixDrones is the first LPPM in this paradigm, to the best of our knowledge. Table 1 shows the summary of the recent studies regarding MZ.

As the recent AI-based techniques enable real-time processing and pattern discovery, traffic analysis has become a powerful feature to support the design of MZ, predicting how, when, and where the mobile nodes will meet the main conditions to perform the mechanism [76, 77, 114, 116, 120]. These factors also turn the MZP dynamic, whose application regions are defined according to the environmental features instead of fixed places, as occurred in different studies [108, 109, 110, 111]. The approaches vary signif-

Table 4.2: Relation of Factors Between MZs for VANETs and for IoD

Factor	VANET	IoD
Mobile node	Car	Drone
Comm. infrastructure	RSU	ZSP
Trajectory “decision”	Driver	ZSP
Traffic Management	Traffic lights, roundabouts, etc.	ZSP
Mix Zone placement	Street intersections	Airway intersections
Coverage area	2D-based	3D-based

icantly, ranging from statistical methods [80, 107] to machine learning models [77, 116]. Some studies also consider group-based solutions, focusing on agreements between the vehicular nodes [75, 76, 112, 113]. Moreover, some methods from the information theory field explore different privacy properties [114, 119, 120].

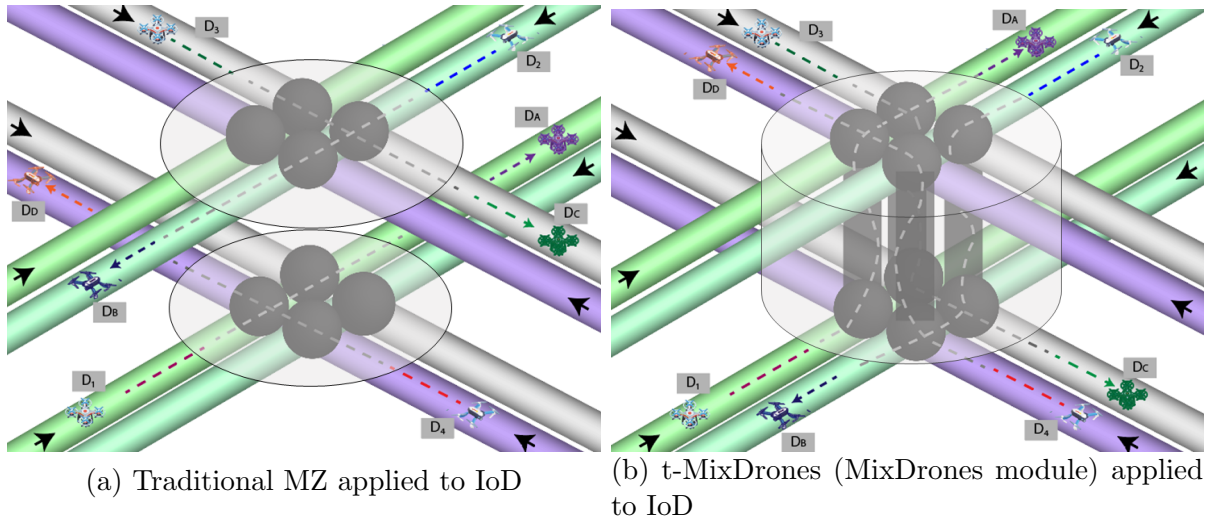
Road intersections represent the central regions to place the MZs since they represent a convergence traffic point with a high probability of vehicle concentration. However, other PoIs have been explored, for instance, parking locations [114, 120], where the elapsed time of a vehicle inside this area is uncertain, enhancing the location privacy provided by the mechanism. Besides these placement factors, several vehicular and roadside features contribute to the design of improved MZ models, mainly the vehicle’s direction, speed, and current location and the monitoring of traffic lights.

The mobility model of nodes in the IoD is significantly different from a traditional mobile network [1]. In VANETs, vehicles move mainly along two axes, the altitude changes smoothly and gradually. On the other hand, civilian drones vary their altitude considerably, performing landings, takeoffs, and even flight deviations due to different factors, such as environment and weather conditions.

This tridimensional mobility becomes an appealing property to explore when we consider the IoD. Figure 4.2a illustrates a scenario where the traditional mechanism is applied with k -anonymity = 4. Airways with the same color represent parallel airways with different altitudes. Black spheres represent the intersection area. The gray ellipses illustrate a placed MZ covering all airways intersections at the same altitude. It is essential to state that the MZ acts as a “black box”. When a drone is inside a MZ, it can not communicate with an LBS sender/user, restricting it to receive/send messages only to ZSP, the trusted party involved.

As we can note, parallel airways remain “isolated” from each other, with the pseudonymization occurring just at the same altitude. Although drones D_1 , D_2 , D_3 and D_4 cross the MZ at a close time, they change only their pseudonym (represented by different arrow colors). Considering an attacker’s point of view, given the trajectories before and after the MZ, the re-identification task of trajectories can be limited at the

Figure 4.2: Comparison of a traditional MZ and t-MixDrones proposal



Source: Elaborated by the author

same airway. Consequently, the de-anonymization of the drone's pseudonym is facilitated, whereas the airway remains the same.

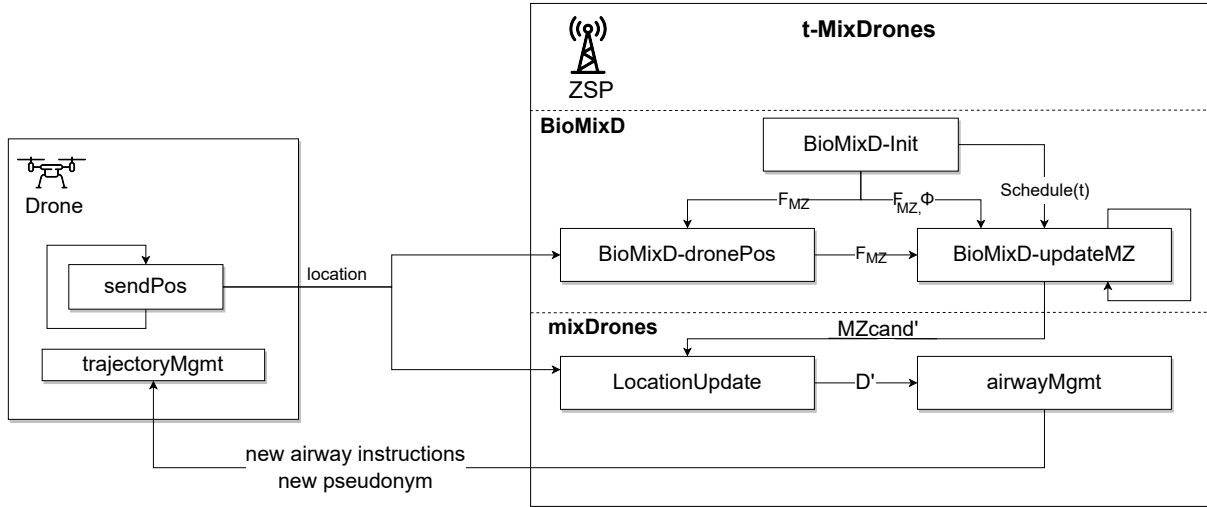
Furthermore, when terrestrial vehicles are inside a MZ, they are typically in constant movement, stopping only for traffic lights, road accidents, or traffic jams. In these scenarios, cars have a similar stationary time interval. On the other hand, airways have no traffic lights, neither are commonly conditioned to traffic jams. Although airway traffic has a better flow compared to urban roads, a large concentration of drones near airway intersections at a close time interval tends to be small. This aspect can diminish the anonymization coverage of drones for configurations requiring many drones inside the application zone to apply the mechanism. Hence, it is necessary to keep the drone inside the MZ for a time interval to decrease the chances of re-anonymize drones.

The t-MixDrones explores these aspects in the design of a new MZ-based mechanism. It embraces a tridimensional and temporal context present in IoD scenarios. MZ regions are logically placed at airway intersections in our design, which is similar to terrestrial crossroads. Hence, we contribute to the advancement of the state of the art regarding location privacy in the IoD environment.

4.2.4 t-MixDrones Architecture

The t-MixDrones is composed of two modules. The first one is the pseudonymization module, named MixDrones, responsible for changing the airway of a drone besides its pseudonym. The second module is the BioMixD, a bio-inspired MZP algorithm that con-

Figure 4.3: Architecture of t-MixDrones



Source: Elaborated by the author

siders the traffic behavior of drones as the main factor in placing the zones. The proposed design of both MixDrones and BioMixD allows them to work cooperatively. Therefore, both strategies compose the architecture of the t-MixDrones. As presented in Figure 4.3, t-MixDrones is a distributed system, being part of the drones and the ZSPs.

The drone's main role is the periodical update regarding its location, which feeds both BioMixD and MixDrones. Besides updating its position, the drone can receive instructions for a new trajectory due to the potential airway change, and a new pseudonym. These data are received and processed in a trajectory management drone's module. Considering the whole scope of t-MixDrones, the drone sustains additional communication with the ZSP as less as possible since it must share its location regardless of the application of t-MixDrones.

The ZSP keeps the most processing effort of t-MixDrones. Considering a real-world application scenario, when the mechanism is deployed in the network, the BioMixD is initialized, scheduling the first call to the `BioMixD-updateMZ` algorithm (described in Section 4.2.6), and spreading the sets $F_{MZ}cand$ and Φ . As the ZSP receives the drone location updates, the `BioMixD-dronePos` algorithm (described in Section 4.2.6) updates the traffic flow counting, which will be considered in the next MZ update.

When BioMixD defines the set $\mathcal{MZ}cand'$, it immediately sends it to the MixDrones module. Thus, when it receives a drone location update, the `locationUpdate` algorithm (Section 4.2.5) verifies if the drone is inside an up-to-date MZ. When the established k -anonymity is satisfied, the `airwayMgmt` algorithm (Section 4.2.5) receives the set of drones \mathcal{D}' that will change the pseudonym, potentially defining a novel airway as a mixing factor. After that, the ZSP sends these data to the preceding drones, completing a cycle of application of t-MixDrones.

In the following, we formally define both MixDrones and BioMixD modules.

4.2.5 MixDrones Module

The MixDrones approach includes airways as a “mixing” factor when the mechanism is applied. The drone can change its airway based on a probability \mathcal{P}_{change} , defining a novel variation of MZ that properly fits into the IoD environment. Our proposal introduces two concepts: Intermediary airways and MixDrones Placement.

► **Intermediary Airways:** it is denoted by a vertical airway connecting two parallel intersections (considering their altitude), allowing a given drone to change its altitude. Moreover, with an airway change, the drone stays longer inside a MZ, contributing to the k -anonymity be satisfied. Differing from common airways, the intermediary airway has a bidirectional flow. Hence, it is mandatory to establish well-defined flight policies to avoid drone collisions.

► **MixDrones Placement:** the MZ are placed at each potential region where drones can be concentrated, grouping airways intersection areas, covering a cylindrical area based on a coverage radius r . The centroid coordinate of the intersections becomes the central point of each MZ for the same altitude.

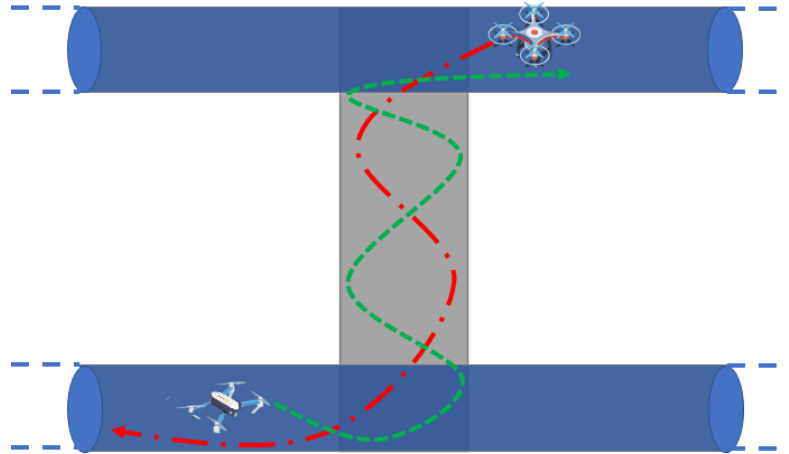
Moreover, given a traffic zone, the ZSPs of this zone are responsible for applying the mechanism, whereas they also manage the drone’s path planning. Figure 4.2b illustrates an example of the MixDrones application in the same scenario of Figure 4.2a, where the vertical black airways represent the intermediary ones. Compared to the traditional approach (Figure 4.2a) every drone changes its airway (\mathcal{P}_{change} is satisfied for all drones in this example) through the intermediary airways, potentially hampering the success of a de-anonymization attack.

The possibility of a drone collision due to airway space competition, as depicted in Figure 4.2b, is an issue that MixDrones considers. For instance, drone D_1 can collide with D_3 or D_4 when it arrives at the upper airway. Hence, the probability of collisions increases, whereas both k -anonymity and the density of drones also increase.

In the literature, different studies proposed drone collision avoidance strategies. Yasin et al. [121] presented an energy-aware and response time minimization method to avoid collisions between drones and obstacles for swarm formations. In this approach, a leader drone detects a given obstacle, calculates its new path planning, and sends the followers’ information. Ahmed et al. [122] also proposed an energy-efficient approach in IoD environments to avoid stationary and mobile obstacle collision. The algorithm is based on the gradient optimization of the drone’s path planning that indicates if there is any available collision-free path. Kumar et al. [123] defined algorithms to avoid collisions in an on-road traffic monitoring scenario. In a drone-centered context, a given drone remains inside a *self-region* with a minimum distance l between any two drones.

Embracing these aspects, ZSPs can coordinate cooperatively the drone’s path plan-

Figure 4.4: Conception of helical mobility to avoid collision between two drones



Source: Elaborated by the author

ning, including speed reduction alerts and redirecting the traffic. Moreover, several drones have sensors to predict and avoid collisions. Nonetheless, in very dense scenarios, more than one drone can share the same intermediary airway. In this case, the ZSP can coordinate the involved drones to perform helical movements in clockwise and/or anticlockwise directions, avoiding their collision, as illustrated in Figure 4.4. Next, we formally define our proposed protocol, including a drone collision avoidance algorithm that handles this issue.

Bearing in mind the previous discussion, we formalize the MixDrones approach. Firstly, it is based on the following assumptions:

- ZSP is the IoD trusty-party that communicates through a reliable communication channel with drones;
- Drones send their position to ZSP constantly, considering a time interval Δ_t , maintaining the management up-to-date;
- Each drone has a group of sensors \mathcal{DH} that can detect obstacles omnidirectionally, considering a distance δ ;
- Given an IoD zone, there is at least one ZSP responsible for managing the airspace;
- In the case of more than one ZSP per zone, there is a “leader” that coordinates the operations.

Formally, the MixDrones approach is defined through two main collaborative algorithms: (i) drone’s location updating, which handles with a drone where it is inside a MZ; and (ii) airways management, which acts directly to process the airway change, selecting intermediary airways to the drones. It is important to state that these algorithms are ZSP-centered, in which it keeps a computational task for each MZ under its responsibility.

Algorithm 1: MixDrones-locationUpdate

```

Input :  $D, Einter, r, k, \mathcal{P}_{change}, d$ 
1  $l \leftarrow \text{predictCurrentLocation}(d.location)$ 
2 if  $isDroneInsideMixZone(R, l)$  then
3   if  $d \notin D$  then
4      $D \leftarrow D \cup \{ \langle d, false \rangle \}$ 
5   if  $|D| \geq k$  then
6      $\langle d, \beta \rangle \leftarrow \text{find tuple of } d \in D$ 
7     if  $\neg \beta$  then
8        $p \leftarrow \text{generatePseudonym}()$ 
9        $\text{send}(d.pseudonym, p)$ 
10       $\beta \leftarrow true$ 
11      $\text{mixDrones-airwayMgmt}(Einter, \mathcal{P}_{change}, d)$ 
12   if  $d \in Einter$  then
13      $\alpha \leftarrow \text{find } \alpha \text{ value of } d \text{ in } Einter$ 
14     if  $\neg isDroneInsideIntermediaryAirway(d, \alpha)$  then
15        $Einter \leftarrow Einter - \{ \langle d, \alpha \rangle \}$ 
16 else if  $d \in D$  then
17    $D \leftarrow D - \{ \langle d, * \rangle \}$ 

```

Before executing these algorithms, an initialization step must be processed aiming to establish the initial configuration of some attributes used by them. All these attributes are initialized as an empty set. They are listed as follows:

- *Einter*: it denotes the intermediary airways as well the drones inside them. Each element of this set is composed of a tuple $\langle d, \alpha \rangle$ in which d is a drone and α is the respective intermediary airway in use;
- *D*: it is a set of drones that are inside a given MZ. Each element is composed of a tuple $\langle d, \beta \rangle$ in which d is a drone and β is a boolean value that indicates if d was already considered to apply the mechanism.

Algorithm 1 presents our proposed mechanism for a specific MZ managed by a ZSP. This mechanism must act as a listener method, being processed at each drone location update. As input, the mechanism requires the sets *Einter* and *D*, the MZ coverage radius r , the level of k -anonymity k , the probability \mathcal{P}_{change} , and the drone d that updates its location.

Given a location update, the mechanism initially predicts the current drone location l (Line 1), whereas there are communication delays involved. This prediction verifies if the drone is inside the MZ based on the coverage radius r (Line 2). If not, it verifies if the drone belongs to the set *D*, indicating that d just left the MZ and then removes it from *D* (Lines 16–17). In an affirmative case, it is necessary to verify three different conditions. The first one is related to the moment that the drone d enters inside the MZ, in which it is added to the set *D* (Lines 3–4).

The second condition verifies if the k -anonymity level is satisfied (Line 5). In an affirmative case, it is necessary to analyze if drone d was already tested, whereas it periodically updates its location (Lines 6–11). If it was not considered yet, a new pseudonym is generated and sent asynchronously to drone d (Lines 8–9), and it is considered as being part of the mechanism’s application (Line 10). Next, the airway management function is called (Line 11), the drone d will change its airway. It is important to note that regardless of the airway change, the drone is always anonymized.

The third condition handles the case of a drone leaving an intermediary airway (Lines 12–15). It is important to state that it requires that the ZSP assigned d to perform an airway change in a previous location update. If the drone d belongs to some intermediary airway α in the set $Einter$, its airway is extracted (Line 13) and, if the drone is no longer inside α , it is removed from $Einter$ (Lines 14–15).

Next, we define and describe the airways management algorithm (Algorithm 2, called in Algorithm 1, Line 11). Firstly, a random probabilistic number is sorted, ranging from the interval between 0 and 1 (Line 1). This number is compared with the \mathcal{P}_{change} , defining if the airway change will be performed (Line 2).

In the affirmative case, the algorithm processes the possible intermediary airways that the drone d can fly during its time inside the MZ (Line 3). With the candidates, the airway will be chosen (Line 4) based on the number of drones flying inside it, i.e., the least congested intermediary airway, which can aid in avoiding the possibility of a drone collision. Hence, for each airway α of the candidate set (Lines 6–10), it is processed the number of drones inside α (Line 7). After, the intermediary airway with lower traffic is allocated to drone d (Lines 8–10). Finally, it is verified if any airway was selected. This verification is necessary since k -anonymity can be addressed when a given drone d is almost leaving it and, consequently, there is no more intermediary airway available to perform the change. If there exists some airway, the drone is associated with α (Line 12), and the ZSP adjusts the path planning of drone d , scheduling the flight over α , where d will flight performing helical movements (Line 13).

4.2.6 BioMixD Module

BioMixD is a bio-inspired and traffic-aware MZP strategy. It considers the traffic behavior of drones as the main factor in placing the zones. As long as drones fly over the airway checkpoints, they deposit pheromones in these regions, which evaporate over time. The checkpoints with more pheromones indicate where the MZ will be placed. Hence, as the traffic behavior changes, the MZ locations also change. BioMixD follows a lightweight

Algorithm 2: MixDrones-airwayMgmt

```

Input :  $E_{inter}, \mathcal{P}_{change}, d$ 
1  $prob \leftarrow \text{randomValue}(\{0, 1\})$ 
2 if  $prob \leq \mathcal{P}_{(change)}$  then
3    $A_{cand} \leftarrow \text{findAirwayCandidates}(E_{inter}, d.pathplanning)$ 
4    $\alpha_{choice} \leftarrow \emptyset$ 
5    $n_{choice} \leftarrow \infty$ 
6   for  $\alpha \in A_{cand}$  do
7      $n_{drones} \leftarrow |\{(*, \alpha)\} \cap E_{inter}|$ 
8     if  $n_{choice} > n_{drones}$  then
9        $\alpha_{choice} \leftarrow \alpha$ 
10       $n_{choice} \leftarrow n_{drones}$ 
11  if  $\alpha_{choice} \neq \emptyset$  then
12     $E_{inter} \leftarrow \langle d, \alpha_{choice} \rangle$ 
13    adjust the path planning of  $d$  with an airway change at  $\alpha$  entrance performing
    helical movements

```

approach to save the drone's battery. Hence, the ZSP is responsible for processing most of the tasks, where the drone updates its position and receives the ZSPs directives. The approach adequately meets the ZSP attributions since both the MZP and anonymization tasks are traffic-based operations.

► **MZP for IoD:** As in the terrestrial mobile networks, the MZP for the IoD is an NP-hard problem. Although it represents a similar optimization task, the MZP definition must be extended for a 3-dimensional environment to meet the IoD characteristics.

- Let \mathcal{CS} be the real-world coordinate system that represents the IoD 3-dimensional environment;
- Let $\zeta = \langle V', r \rangle$ be an MZ region candidate, composed of a set $V' \subset G.V$, representing the intersection nodes related to the MZ region, and a radius r that indicates the MZ coverage range;
- Let \mathcal{MZcand} be the set of all the possible MZ candidates, such that $\mathcal{MZcand} = \{\zeta_0, \dots, \zeta_i\}$, for i possible MZ regions;
- Let $\mathcal{MZcand}' \subset \mathcal{MZcand}$ be the subset of the placed MZ regions;
- Let \mathcal{M}_p be a set of metrics related to the MZP problem, for instance, the number of drones that traverse a given ζ ;
- Let n be the max number of placed MZ allowed in the environment;

Considering this system model, the MZP in IoD can be stated by the following question: “Given \mathcal{MZcand} MZ candidates in the IoD environment G , what are the best

set of candidates $\mathcal{MZ}cand'$ such that $\mathcal{MZ}cand'$ provides the best result when applied to \mathcal{M}_p ”?

Mathematically, MZP for the IoD can be defined by the following optimization problem:

$$\max_{m \in \mathcal{M}_p} \sum_{m \in \mathcal{M}_p} f_m(\mathcal{MZ}cand', D), \forall \mathcal{MZ}cand' \subset \mathcal{MZ}cand \quad (4.1)$$

Subject to: $|\mathcal{MZ}cand'| \leq n$

The optimal MZP is given by the highest sum of the results obtained by the application f of each metric $m \in \mathcal{M}_p$ for all the subsets of MZ regions with n regions at most. The subset $\mathcal{MZ}cand'$ associated with the highest sum represents the best regions to place an MZ. Whereas the number of subsets $\mathcal{MZ}cand'$ increases, the computational effort to process the function also increases since the subsets combination has a factorial order [110]. Thus, it is necessary to propose strategies that balance a good solution for MZP and the computational effort.

► **BioMixD Main Concepts:** Over the years, bio-inspired algorithms have been massively applied as heuristics to provide accurate results in optimization problems. These algorithms have an “intelligent behavior” since they are designed based on biological organisms and systems, being self-adaptive. Neural Networks (NN), Genetic Programming (GP), Ant Colony Optimization (ACO), and Particle Swarm Optimization (PSO) are well-known bio-inspired algorithms [124]. These last two also embrace a bio-inspired category named “swarm intelligent” algorithms. Shortly, they consider that individual agents interact with the entire system, providing and receiving feedback for their actions, leading to a collective decision.

ACO [125], specifically, generates optimized solutions based on the premise that the agents deposit pheromones while walking. With time, the path with more pheromones induces other agents to follow it. As time goes by, the pheromone evaporates from non-visited regions. ACO has been designed as a solution for different tasks of ITS, such as the vehicle job assignment problem [126], and trajectory decision [127]. To the best of our knowledge, ACO has never been designed as a solution for MZP. However, its main concepts fit into the IoD environment considering a drone’s traffic point of view.

Table 4.3 shows the mapping of the main concepts between a traditional ACO and the BioMixD. In the BioMixD, the drone represents the agent that traverses the available paths, carrying and depositing the corresponding pheromone, here represented as the traffic flow through an MZ region candidate ζ . In other words, when a given drone $d \in D$ flies over a candidate ζ , d deposits a “pheromone unit” in ζ , which will evaporate over time, representing the evaporation factor.

A significant conceptual difference between ACO and BioMixD is that the drone does not follow the path with more pheromones, keeping the path planning provided by

Table 4.3: Mapping of the main concepts of ACO and BioMixD

Concept	ACO	BioMixD
Agent	Ant	$d \in \mathcal{D}$
Feedback factor	Pheromone	Traffic flow through ζ
Pherom. deposition	Ant's path	ζ
Solution selection	By epoch	By a time window Δ_t
Goal	Best path discovery	Best $\mathcal{MZ}cand'$

the ZSP. Thus, drones can be considered “independent” agents. On the other hand, MZs are placed according to the pheromone rate. Based on a time window Δ_t and in a maximum number n of MZs to be placed, the regions with a higher pheromone rate form the $\mathcal{MZ}cand'$ set, representing where MZ will be located until the next time window Δ_t ends.

The pheromone rate ϕ_ζ of a given MZ region ζ over a time window Δ_t is given by Equation 4.2, where ρ is the pheromone's evaporation rate, and f is a function that calculates the rate of drones that traverses the MZ region ζ during Δ_t . f is described in Equation 4.3.

$$\phi_\zeta(\Delta_t) = (1 - \rho) \phi_\zeta(\Delta_{t-1}) + f(\zeta, \mathcal{D}, \Delta_t) \quad (4.2)$$

$$f(\zeta, \mathcal{D}, \Delta_t) = \frac{\sum_{d \in \mathcal{D}} \begin{cases} 1, & \text{if } d \text{ traverses } \zeta \text{ during } \Delta_t \\ 0, & \text{otherwise} \end{cases}}{|\mathcal{D}|} \quad (4.3)$$

Summarily, regions with a low pheromone tend to keep a low value unless a greater number of drones fly over there over time. Likewise, regions with a higher pheromone tend to keep a high value unless a few drones fly over there along the time. This property ensures that the placed \mathcal{Z}' reflects not just the traffic at a given time interval but a recent traffic history.

BioMixD is defined through three complementary distributed algorithms. Algorithm 3 describes how to initialize the mechanism. Algorithm 4 presents the drone position update, and Algorithm 5 refers to the MZ placement update. We describe them in detail and discuss their time complexity as follows.

► **Algorithm 3 BioMixD Initialization** : Before applying the t-MixDrones mechanism in a given environment, the BioMixD must have an initial information about the drones traffic flow during a time interval $\langle 0, t_{init} \rangle$. To address this information, it uses a data structure $F_{\mathcal{MZ}}$ such that each element $f_z \in F_{\mathcal{MZ}}$ is composed of a tuple $\langle \zeta, i \rangle$ where $\zeta \in \mathcal{MZ}cand$ and $i \in \mathbb{Z}$, representing that i drones fly over the MZ candidate ζ . Moreover, the initial pheromone value ϕ of each $\zeta \in \mathcal{MZ}cand$ must be initialized. Thus,

Algorithm 3: BioMixD-Init

Input : $\mathcal{MZ}cand, t_{init}, \Phi$
Output: $F_{\mathcal{MZ}}, \Phi$

- 1 $F_{\mathcal{MZ}} \leftarrow \emptyset$
- 2 $\Phi \leftarrow \emptyset$
- 3 **for** $\zeta \in \mathcal{MZ}cand$ **do**
- 4 $F_{\mathcal{MZ}} \leftarrow F_{\mathcal{MZ}} \cup \{\langle \zeta, 0 \rangle\}$
- 5 $\Phi \leftarrow \Phi \cup \{\langle \zeta, 1 \rangle\}$
- 6 $t \leftarrow$ current system time
- 7 Schedule an event call for **BioMixD-UpdateMZ** at time $t + t_{init}$

BioMixD considers a similar data structure Φ to keep these values. Each $\phi \in \Phi$ is also composed of a tuple $\langle \zeta, p \rangle$ such that $\zeta \in \mathcal{MZ}cand$ and $p \in \mathbb{R}$, representing the current pheromone value of the MZ candidate ζ .

Both structures carry the main data of BioMixD, being accessed and modified through all BioMixD algorithms. Therefore, they must be modeled as a data structure with rapid access and update, for instance, a binary heap or a hash-based structure. Considering that each ζ has a unique coordinate, it can be coded as a key to identify each region in both $F_{\mathcal{MZ}}$ and Φ . In this work, we design both structures as two distinct binary heaps whose key is coded by ζ 's coordinate. However, other arrangements can be considered, and our proposal is not limited nor dependent on this structure.

As input, Algorithm 3 requires the set of MZ candidates $\mathcal{MZ}cand$, the initial learning time interval t_{init} , and the pheromone set Φ . Initially, the algorithm initializes both sets as empty (Lines 1-2). After, each corresponding MZ candidate $\zeta \in \mathcal{MZ}cand$ has an associated tuple in the sets $F_{\mathcal{MZ}}$ and Φ (Lines 3-5). The first is initialized with 0 since any drone already flies over the region, and the second is initialized with 1 as the initial pheromone value. Lastly, the algorithm appropriately schedules an event call to Algorithm 5 after the time interval t_{init} , when the MZP will start to occur (Lines 6-7). It gives the BioMixD time to gather initial information about the traffic flow.

► **Algorithm 3 BioMixD Time Complexity Analysis** The initialization of the sets $F_{\mathcal{MZ}}$ and Φ as empty sets (Lines 1-2) as well as the schedule of a call for Algorithm 5 (Lines 6-7) occurs in $\Theta(1)$ since they are addressed through well-known mathematical set operations and system calls, respectively. On the other hand, the iteration regarding the flow and pheromone of each MZ candidate (Lines 3-5) demands an ordered insertion based on the ζ key to facilitate its access, as previously discussed. As we consider the structures a binary heap, the insertion has a time complexity of $\mathcal{O}(\log |\mathcal{MZ}cand|)$, occurring $|\mathcal{MZ}cand|$ times for each structure. Hence, the time complexity can be defined as follows:

$$\begin{aligned}
T(\text{BioMixD-Init}) &= 4\Theta(1) + 2|\mathcal{MZ}cand| \mathcal{O}(\log |\mathcal{MZ}cand|) \\
&= \mathcal{O}(|\mathcal{MZ}cand| \log |\mathcal{MZ}cand|)
\end{aligned} \tag{4.4}$$

Algorithm 4: BioMixD-DronePos

Input : $d_{pos}, \mathcal{MZ}cand, F_{\mathcal{MZ}}, r$
Output: $F_{\mathcal{MZ}}$

- 1 **for** $\zeta \in \mathcal{MZ}cand$ **do**
- 2 **if** $verifyDroneInsideMZ(d_{pos}, \zeta, r)$ **then**
- 3 $\langle \zeta, i \rangle \leftarrow search(\zeta, F_{\mathcal{MZ}})$
- 4 $i \leftarrow i + 1$
- 5 $update(\langle \zeta, i \rangle, F_{\mathcal{MZ}})$

► **Algorithm 4 BioMixD Drone Position Update** : The algorithm is called every time a drone d updates its position d_{pos} , sending it to the nearest ZSP. Its main objective is to update the $F_{\mathcal{MZ}}$ of an MZ candidate ζ since d flies near ζ . The algorithm requires as input the drone's position D_{pos} , the set of MZ candidates $\mathcal{MZ}cand$, the traffic flow structure $F_{\mathcal{MZ}}$, and the coverage radius r that indicates if a given drone is close enough to the MZ. Algorithm 4 consists of iterating over the MZ candidates and verifying if the drone is near a given candidate ζ based on the radius r (Lines 1–5). In the affirmative case, the tuple related to ζ is recovered on $F_{\mathcal{MZ}}$ (Line 3), increased (Line 4), and updated (Line 5).

► **Algorithm 4 BioMixD Time Complexity Analysis** : Given the drone position update request, the time complexity depends on the search (Line 3) and update (Line 5) operations, according to the data structure model. Using binary heap structures, the search has a time complexity of $\mathcal{O}(\log |F_{\mathcal{MZ}}|)$. As the update operation does not change the key coding of the tuple, it is not necessary to rebuild the heap structure. Considering that the sets $\mathcal{MZ}cand$ and $F_{\mathcal{MZ}}$ have the same size since they refer to the MZ region candidates, the time complexity can be defined as follows:

$$\begin{aligned} T(\text{BioMixD-DronePos}) &= |\mathcal{MZ}cand| \mathcal{O}(\log |F_{\mathcal{MZ}}|) \\ &= \mathcal{O}(|\mathcal{MZ}cand| \log |\mathcal{MZ}cand|) \end{aligned} \quad (4.5)$$

► **Algorithm 5 BioMixD Update Mix Zones** : After the first scheduling generated by Algorithm 3, the MZ update occurs periodically given a time interval Δ_t . Besides this time interval, the algorithm requires as input the sets $\mathcal{MZ}cand$, $F_{\mathcal{MZ}}$, Φ , and the total number n of MZ to place. As auxiliary structures, we have a counter η that stores the number of times the drones flew over the MZ candidates, and the set $\mathcal{MZ}cand'$ that stores the selected MZs. Initially, they are assigned as 0 and \emptyset , respectively (Lines 1-2).

After, the algorithm calculates the number of times that the drones flew over the MZ candidates. This task is addressed by iteration over all the tuples of the candidates $\zeta \in F_{\mathcal{MZ}}$, increasing the value of η (Lines 3-5). This sum properly represents the function f defined in Equation 4.3, which is applied as a part of the pheromone update function.

Algorithm 5: BioMixD-UpdateMZ

Input : $\mathcal{MZcand}, F_{\mathcal{MZ}}, \Phi, \Delta_t, n$
Output: $F_{\mathcal{MZ}}, \Phi, \mathcal{MZcand}'$

- 1 $\eta \leftarrow 0$
- 2 $\mathcal{Z}' \leftarrow \emptyset$
- 3 **for** $\zeta \in \mathcal{MZcand}$ **do**
- 4 $\langle \zeta, i \rangle \leftarrow$ find the tuple related to ζ in $F_{\mathcal{MZ}}$
- 5 $\eta \leftarrow \eta + i$
- 6 **for** $\zeta \in \mathcal{MZcand}$ **do**
- 7 $\langle \zeta, i \rangle \leftarrow$ find the tuple related to ζ in $F_{\mathcal{MZ}}$
- 8 $\langle \zeta, \phi \rangle \leftarrow$ find the tuple related to ζ in Φ
- 9 $\phi \leftarrow$ update ϕ according Eq. 4.2 considering that $\frac{i}{\eta}$ represent the function f in Eq. 4.3
- 10 update the tuple related to ζ in Φ with the values $\langle \zeta, \phi \rangle$
- 11 $\Phi' \leftarrow$ extract the n tuples with higher ϕ
- 12 **for** $\langle \zeta, \phi \rangle \in \Phi'$ **do**
- 13 $\mathcal{MZcand}' \leftarrow \mathcal{MZcand}' \cup \{\zeta\}$
- 14 **for** $\zeta \in \mathcal{MZcand}$ **do**
- 15 $\langle \zeta, i \rangle \leftarrow$ find the tuple related to ζ in $F_{\mathcal{MZ}}$
- 16 $i \leftarrow 0$
- 17 update the tuple related to ζ in $F_{\mathcal{MZ}}$ with the values $\langle \zeta, i \rangle$
- 18 $t \leftarrow$ current system time
- 19 Schedule an event call for **BioMixD-UpdateMZ** at time $t + \Delta_t$

These updates occur for each MZ candidate ζ (Lines 6-10), considering the stored traffic flow in its region (Line 7) and its current pheromone (Line 8). Based on all these data, the algorithm calculates the new pheromone value according to Equation 4.2 (Line 9), updating it in the set Φ (Line 10).

The next step is obtaining the n tuples from Φ with the highest pheromone, representing the MZ regions with more increased traffic flow (Line 11). Thus, the set \mathcal{MZcand}' includes the regions ζ associated with these pheromones (Lines 12-13). As the historical traffic flow behavior is inherent to the pheromone updates, BioMixD does not consider a historical counting of the traffic flow of each ζ . Thus, they need to be re-initialized (Lines 14-17). Lastly, a new update of MZ is scheduled based on the time interval Δ_t .

► **Algorithm 5 (BioMixD) Time Complexity Analysis** Similar to Algorithm 3, the initialization of data structures (Lines 1-2) as well as the system schedule (Lines 18-19) has a time complexity of $\Theta(1)$. The iteration of Lines 3-5 is dominated by the complexity of the **search** method, being $\mathcal{O}(\log |\mathcal{MZcand}|)$, as presented in the analysis of Algorithm 4. Likewise, the iterations of Lines 6-10 and Lines 14-17 have the same time complexity since they have **search** and **update** operations. The extraction of the n highest pheromones (Line 11) demands the creation of an auxiliary structure sorted by the pheromones. It is necessary because the MZ location sorts the binary heap Φ instead of the pheromone

value. Hence, it is possible to use an auxiliary binary heap whose creation has a time complexity of $\mathcal{O}(|\mathcal{MZ}cand| \log |\mathcal{MZ}cand|)$. The selection of the best ζ in Lines 12-13 has a complexity $\Theta(n)$. However, $|\mathcal{MZ}cand|$ will be greater than or equal to n since we can not place more MZ than the available. Hence, the complete time complexity can be defined as follows:

$$\begin{aligned} T(\text{BioMixD} - \text{UpdateMZ}) &= 4\Theta(1) + 4\mathcal{O}(|\mathcal{MZ}cand| \log |\mathcal{MZ}cand|) + \Theta(n) \\ &= \mathcal{O}(|\mathcal{MZ}cand| \log |\mathcal{MZ}cand|) \end{aligned} \quad (4.6)$$

4.2.7 Simulation Setup and Performance Evaluation

This section describes our experimental evaluation of t-MixDrones through extensive simulations. With this evaluation, our goals are threefold: (i) analyze how comprehensive t-MixDrones is in terms of location privacy; (ii) investigate how resilient t-MixDrones is when facing de-anonymization attacks; and (iii) verify at what configurations and in what level the BioMixD contributes to the location privacy provided by t-MixDrones.

► **Metrics:** Our experimental evaluation intends to investigate the performance of t-MixDrones as an LPPM in several IoD configuration scenarios (discussed in the next section), comparing it with a traditional MZ mechanism (applied for VANETs), and with a version with MixDrones, only. In this second case, our goal is to evaluate in what level the BioMixD approach contributes to the mechanism as a whole.

We applied four different metrics to evaluate them: Coverage Rate (C_{rate}), Re-anonymization Average Rate (RAR), Trajectory Matching Accuracy (TMA), and the Increasing Travel Time Rate ($ITTR$). They are described as follows.

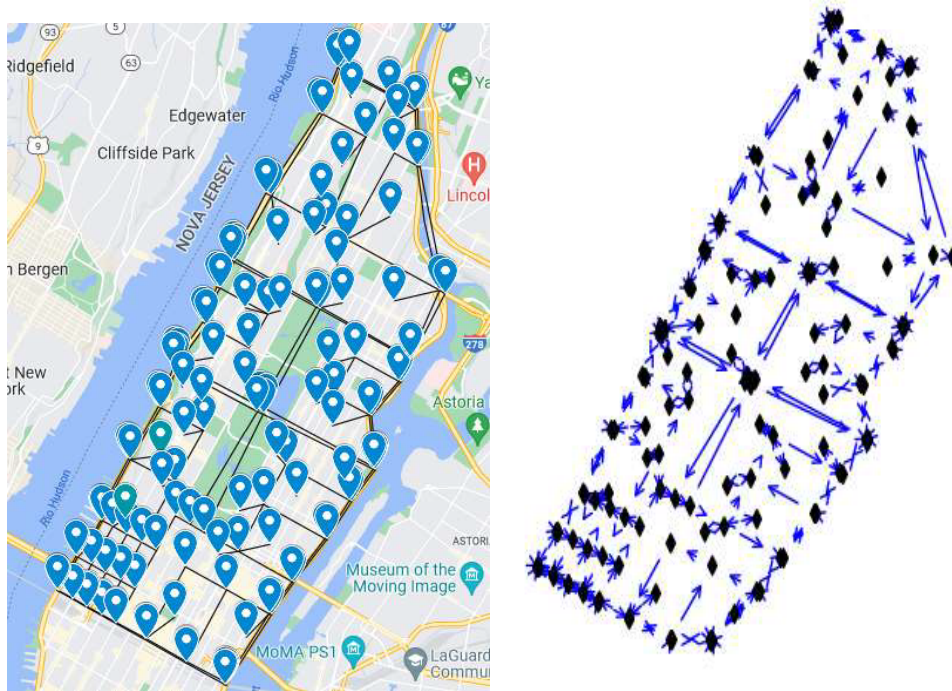
- Coverage Rate (C_{rate}): this metric measures the fraction of drones that are anonymized at least once in the network. Let D be the total number of drones, and $\beta_i \in \{0, 1\}$ a boolean value that indicates if the drone i was anonymized. Thus,

$$C_{rate} = \frac{\sum_{i=0}^D \beta_i}{D} \quad (4.7)$$

- Re-anonymization Average Rate (RAR): this metric calculates how many times a drone is re-anonymized per trip. Equation 4.8 defines it, where d represents a given drone, \mathcal{T} is the set of performed trips, and `anonymizations` is a function that returns the number of anonymizations of the drone d during the trip.

$$RAR = \frac{\sum_{trip=1}^{|\mathcal{T}(d)|} \text{anonymizations}(d, \text{trip})}{|\mathcal{T}(d)|} \quad (4.8)$$

Figure 4.5: Simulated urban scenario for t-MixDrones performance evaluation



(a) Map of Manhattan Island, NY. (b) Topological graph representation. Black line segments represent the corresponding public and flyable airspace. Blue pins represent the available POIs. Black dots indicate a node, and the blue arrows represent the directed edges.

Source: Elaborated by the author

- Trajectory Matching Accuracy (TMA): measures the rate of correct re-identification of trajectories, considering the attacker's perspective. The greater the TMA , the lower the t-MixDrones resilience. Let N_{reid} be the number of trajectories correctly re-identified, and $|T_p|$ the sum of related trajectories before and after the Mix Zone. Thus, TMA can be defined as follows.

$$TMA = \frac{N_{reid}}{|T_p|} \quad (4.9)$$

- Increasing Travel Time Rate ($ITTR$): based on the travel time of the original trip – denoted by T_{origin} – it measures the rate of increasing time T_{mix} generated by the mechanism application.

$$ITTR = \frac{T_{mix}}{T_{origin}} \quad (4.10)$$

► **Environment simulation:** We model a region of Manhattan Island, NY, as the urban scenario to support these assumptions to evaluate t-MixDrones. Figure 4.5a shows the geographic map of this scenario, covering an area of around 39.8 km². As we observe

on the map, the drones are not allowed to fly over all the airspace, following well-defined airway boundaries. Likewise, drones can take off/land only in specific locations, as indicated by the blue pins. The IoD network system models this urban environment as a graph representation (Figure 4.5b), where each node has an associated localization, and the edges have a specific direction.

We carried out extensive simulations to evaluate the t-MixDrones. We used the IoDSim¹ as a simulation tool. It is an IoD simulation environment integrated with the INET framework through the simulator OMNeT++. The IoD concept used in the simulations follows the layered network proposed by Gharibi et al. [5], where drones fly over well-defined airways in which the ZSP manages the airspace. We model an urban environment scenario that represents a piece of Manhattan Island. Table 4.4 shows a list of relevant parameters of the simulation environment. A parameter with a bold font indicates that its value varies, representing different configurations.

We define these parameters considering a reasonable time interval to allow a drone to communicate and include the mechanisms. During the three hours of simulation, the drones fly over the airways following a previously defined path planning, with scheduled landings and takeoffs. To address a thorough analysis, we compare three LPPMs: the traditional MZ scheme, commonly applied in vehicular networks; the MixDrones, solely (the MZ regions are defined *a priori*); and t-MixDrones. For each mechanism, we set 72 configurations, combining the different parameterizations of the number of airways, the number of MZ to place, the coverage radius, and the k -anonymity. Regarding the communication model, IoD nodes have the same protocols and radio configurations, using TCP, AODV, and CSMA/CA. The radio operates following the standard 802.11n with modulation mode of 1×1 20 MHz, having a maximum throughput of 72.2 Mbps.

Furthermore, we execute 30 simulations with different seeds for each configuration, replicating the same scenario for each mechanism. Hence, we perform 6,480 simulation experiments, highlighting our approach’s robustness and statistical validation, leading to results with a 95% confidence interval.

4.2.8 Results and Discussion

This section presents the results of our experimental evaluation, considering the discussed metrics. We conducted a thorough analysis of each one, highlighting the performance of t-MixDrones and comparing it with both the traditional Mix Zones for grounded vehicular networks and MixDrones protocol, solely. It is essential to state that the in-

¹<https://iodsim.manna.team>

Table 4.4: Simulation parameters regarding the t-MixDrones performance evaluation

Parameter	Value
<i>General Parameters</i>	
Simulation time	180 minutes
Environment boundaries	$3.8 \times 10.5 \times 0.25 \text{ km}^3$
Drone speed	uniform [36–54] km/h
Mobility pattern	Gauss-Markov
#Drones	50
#Airways	{2,4}
Mechanisms	{traditional MZ, MixDrones, t-MixDrones}
<i>t-MixDrones Parameters</i>	
Δ_t	1 min
ρ	0.1
\mathcal{P}_{change}	0.8
#Mix Zones to place	{1,2,4}
Mix Zones coverage radius	{50 m, 250 m}
k -anonymity	{2, 4, 6}

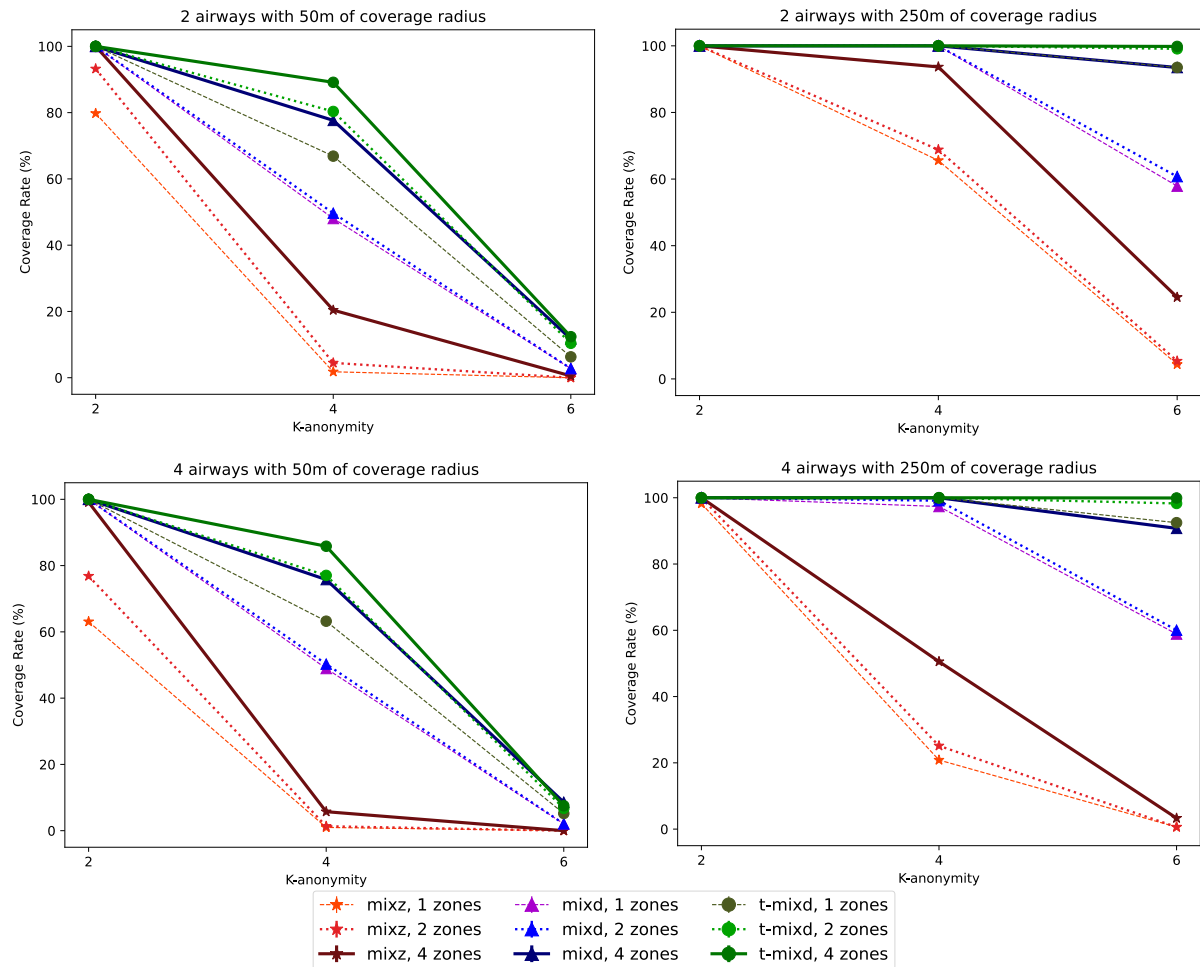
terval errors for all results are less than 1% and, therefore, they are not represented in the charts. Lastly, we present a general discussion considering the results from a broad perspective.

► **Coverage Rate results:** The C_{rate} evaluates a fundamental privacy aspect since it measures how many drones have at least one anonymization. This metric shows the rate of the network nodes that have protection against trajectory-based location attacks. Figure 4.6 shows the results of C_{rate} . Each chart presents the evaluated mechanisms' results with a different number of MZ placed over all the k -anonymity values, varying the number of available airways and the coverage radius. In the left column, we have the configurations with 50 m of coverage radius, and in the right column, the configurations with 250 m. In the upper row, the configurations with two airways, and in the lower row, the configurations with four airways.

It is noteworthy that the t-MixDrones mechanism achieves the high C_{rate} in all scenarios, considering all numbers of placed MZ. For a higher radius coverage, when the value of k -anonymity is higher, t-MixDrones keeps covering around 100% of the drones, regardless of the number of airways, whereas MixDrones and the traditional MZ decrease the rate significantly. This result indicates that the dynamic MZP performed by the BioMixD approach plays a crucial role in drone coverage through t-MixDrones.

As expected, the MixDrones-based mechanisms have a higher performance than the traditional MZ mechanism, reinforcing that the model of MZ as a cylindrical region represents better the IoD characteristics, providing higher privacy. Another insight about the C_{rate} is that the number of available airways does not affect MixDrones and t-MixDrones performance. The cylindrical model is also responsible for this performance,

Figure 4.6: Results of Coverage Rate regarding the t-MixDrones performance evaluation



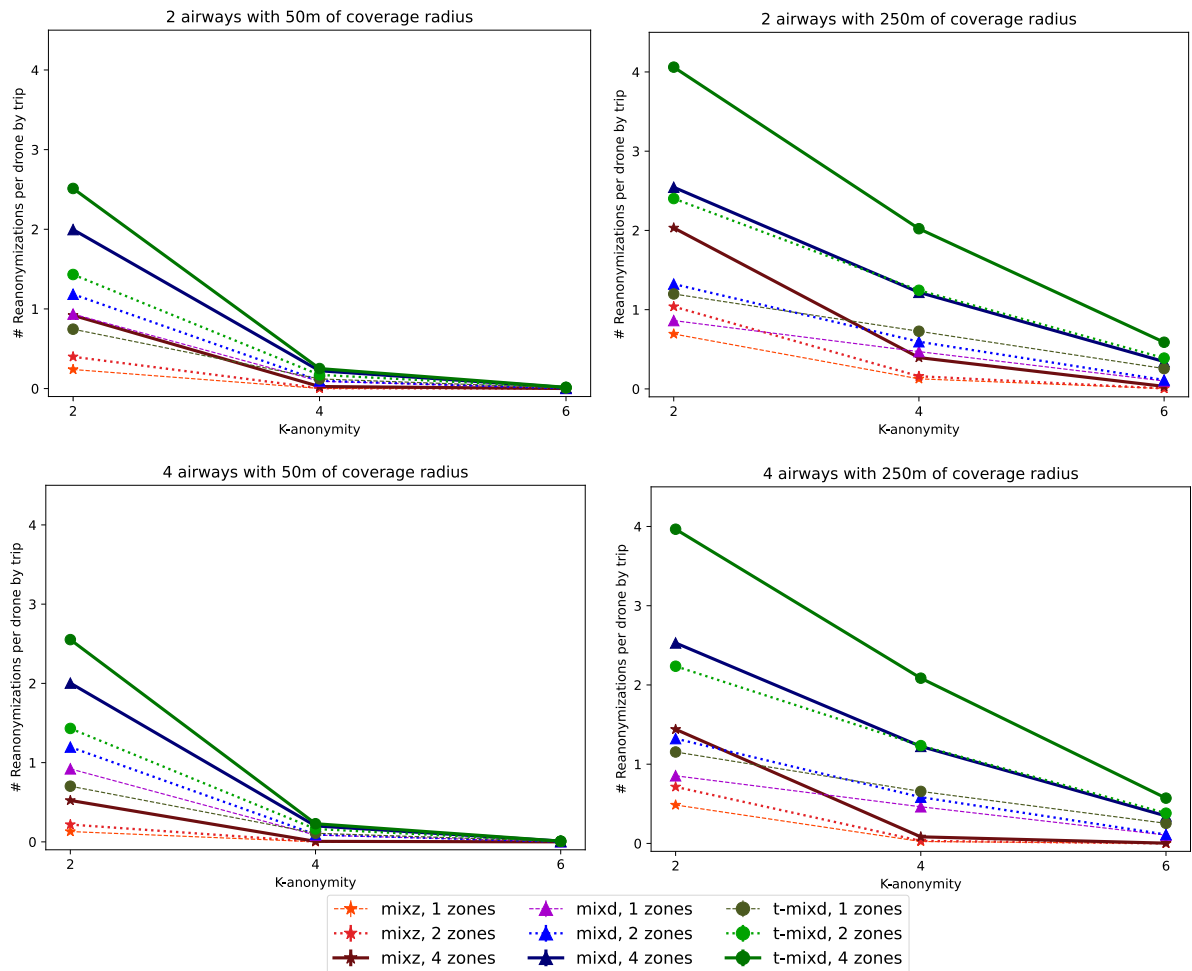
Source: Elaborated by the author

covering all airway altitudes. Regardless of how the drones are spread out at different altitudes, the engine will always capture them as being in the same region.

However, the results point out a challenge that demands further investigation. For a low coverage radius and a higher value of k -anonymity, all the mechanisms provide a coverage rate lower than 20%. Considering the dimensions of the simulated environment and its airways topology, the conditions that satisfy the application of the mechanism are very restricted. Hence, the coverage radius must be tuned accordingly before deploying the mechanism.

► **Results of Re-anonymization Average Rate:** The C_{rate} indicates how many drones are covered by the LPPM, but it does not consider how frequently drones are anonymized. In this case, the RAR metric provides this overview, as depicted in Figure 4.7. The charts follow the same layout as Figure 4.6, exhibiting the three evaluated LPPMs. Each value represents the average of how many times a given drone has been anonymized during a single trip.

Figure 4.7: Results of Re-anonymization Average Rate regarding the t-MixDrones performance evaluation

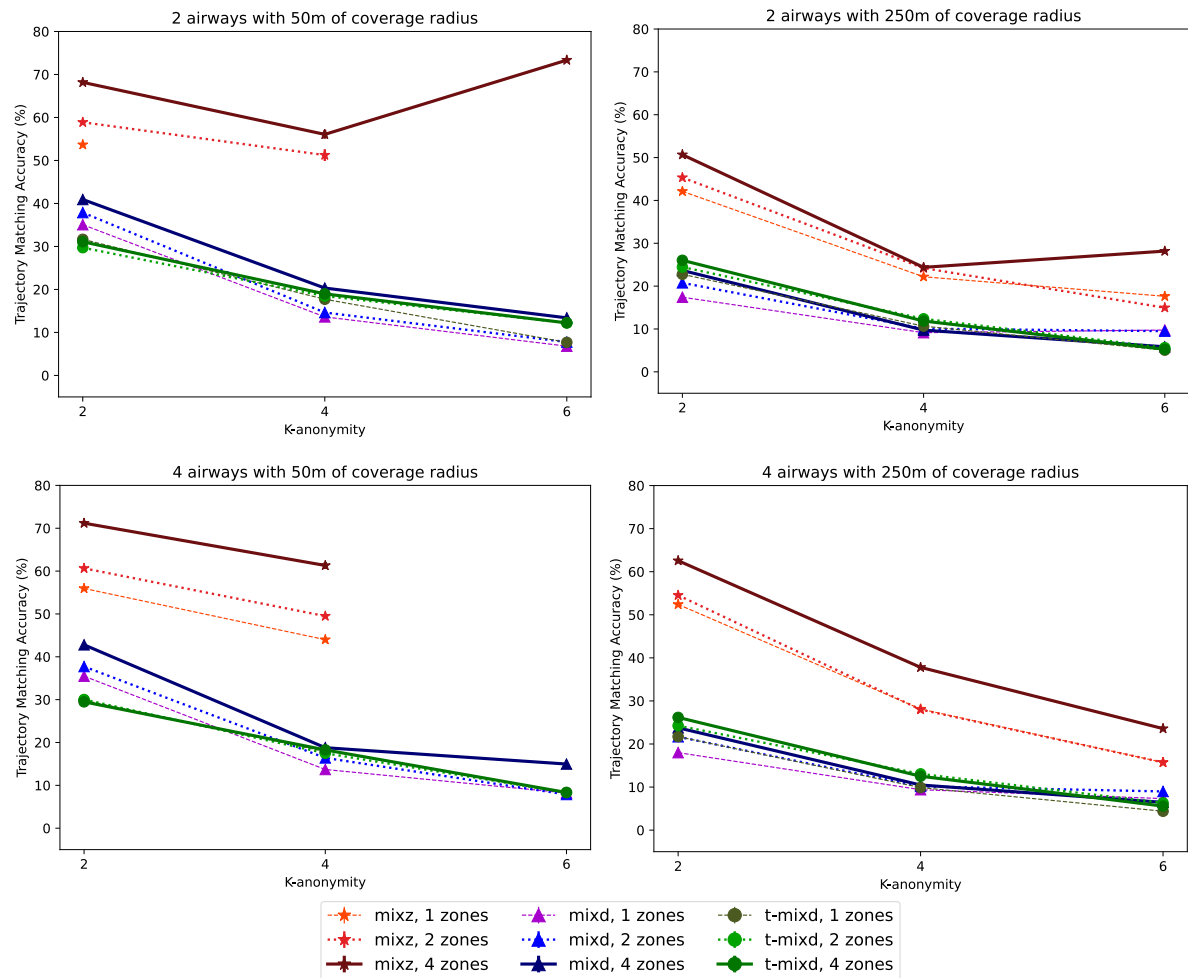


Source: Elaborated by the author

As in the C_{rate} metric, the t-MixDrones mechanism provides higher levels of re-anonymization in all configurations. Furthermore, the higher the number of MZ placed, the higher the RAR for t-MixDrones. Although t-MixDrones prevails in all configurations, the number of MZ placed is a major aspect of providing a higher frequency of re-anonymizations. For instance, the MixDrones mechanism with four placed MZ provides a higher RAR than the t-MixDrones with two placed MZ. Nonetheless, as the required k -anonymity increases, the difference between these two configurations decreases, highlighting again that, for more restricted environments, t-MixDrones presents better performance.

The value of coverage radius also represents a factor of impact to provide a higher re-anonymization per trip. The radius of 250 m provides almost a double RAR value compared to the radius of 50 m. Nevertheless, regardless of the configuration, a higher value of k -anonymity implies few re-anonymizations, even for the higher coverage radius. When $k = 6$, for instance, the average re-anonymization is less than 1, indicating that

Figure 4.8: Results of Trajectory Matching Accuracy regarding the t-MixDrones performance evaluation



Source: Elaborated by the author

there are trips where the drone does not change its pseudonym.

► **Results of Trajectory Matching Accuracy:** From a location privacy point of view, besides a high rate of both coverage and frequency of re-anonymizations, an LPPM must have proper resilience against location privacy attacks. In this study, *TMA* provides this analysis, where a high *TMA* points out that an LPPM provides a lower level of location privacy. Figure 4.8 shows the results regarding *TMA*. The charts are presented in the same way as in the former metrics. It is important to note that, for these charts, a lower value represents better performance. Moreover, the charts present the results from the configurations that performed at least one anonymization. For instance, in the configuration with four airways and 50 m of coverage radius, the traditional MZ mechanism does not perform any anonymizations when k -anonymity = 6.

The results point out that both t-MixDrones and MixDrones present an outstanding resilience compared to the traditional MZ. As the value of k -anonymity increases, the

resilience facing the attack also increases. In the more restricted scenarios, the attacker links correctly around 10% of the trajectories, representing a noteworthy location privacy protection from these two mechanisms.

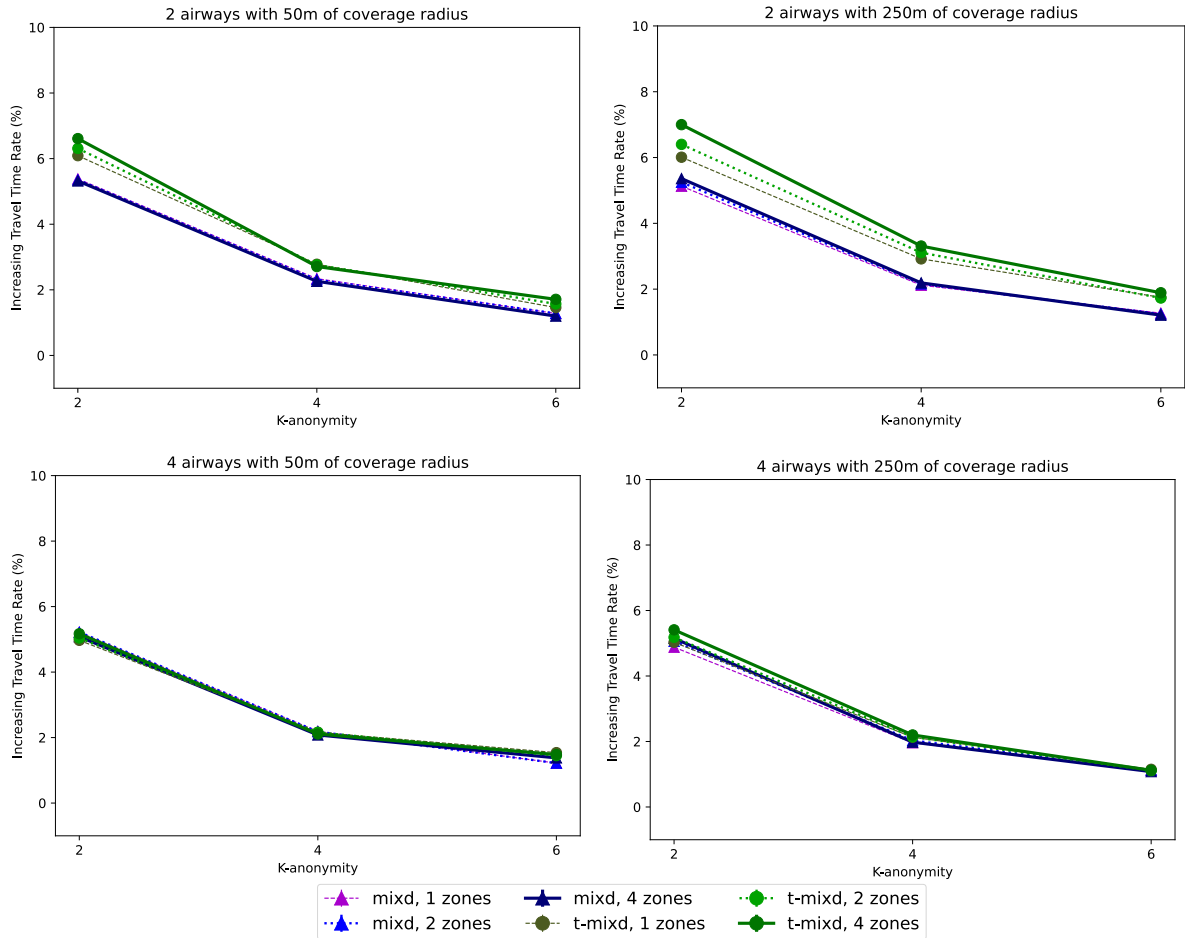
Differing from C_{rate} and RAR , t-MixDrones and MixDrones present a similar performance considering the overall evaluation. For the configurations with 50 m, the t-MixDrones presents a better performance with lower k -anonymity values, but MixDrones overcomes t-Mixdrones for a higher k . On the other hand, the opposite happens for a coverage radius of 250 m, even with a smaller performance difference. This similar performance occurs due to the airway change representing the main “mixing” factor that confounds the attacker. Considering that both MixDrones and t-MixDrones have the same architecture regarding the performance of airway change, it is quite reasonable that they have similar resilience. Furthermore, we can observe that the t-MixDrones results present behavior that approaches a linear function in the order of the value of k -anonymity. It can be strictly related to the higher C_{rate} and RAR provided by this mechanism. Indeed, less coverage and less anonymization lead to outliers in TMA , as we can observe with the traditional MZ mechanism.

► **Results of Increasing Travel Time Rate:** Both t-MixDrones and MixDrones request that drones change their airways, and, thus, there is a potential time travel delay caused by the airway change. Depending on the provided service, delays can cause severe complications in the QoS of a given drone. Figure 4.9 presents the results of $ITTR$ for these two mechanisms. In this metric, we take the time travel of the traditional MZ mechanism as the baseline since the mechanism does not perform any airway change and, therefore, reflects the drone’s time travels without delays.

Both mechanisms cause an $ITTR$ of less than 8% of the baseline travel for all configurations. For configurations with fewer available airways, t-MixDrones presents a slightly higher delay, explained by the greater distance between the two available airways. Furthermore, since t-MixDrones perform a higher number of anonymizations, the travel time increases in these scenarios. The k -anonymity also influences the travel time. As we already discussed, higher values of k set a restricted scenario for applying the mechanism. Hence, as the anonymizations – and consequently the airway change – occur less, the $ITTR$ diminishes.

In our experiments, a drone performs trips that take around 15 minutes. Hence, the delay corresponds to less than 1.5 minutes, on average. The impact of the delay is fundamentally related to the type of service. For instance, emergency tasks can act in real time and do not tolerate significant delays. Furthermore, the distance between the current and the new airways and the drones’ speed can influence the delay caused by the change of airways. These issues require a thorough analysis of the following aspects: (i) the provided drone’s service; (ii) the arrangement of the airways; and (iii) how the speed

Figure 4.9: Results of Increasing Travel Time Rate regarding the t-MixDrones performance evaluation



Source: Elaborated by the author

of drones can affect the delay and continue to guarantee a low *ITTR* for the evaluated mechanisms.

► **General Discussion:** Through this evaluation, we can observe that t-MixDrones provides the best anonymization coverage for drones in all evaluated scenarios, especially when the MZ coverage radius is broader. In these scenarios, t-MixDrones provides more than 95% of coverage regardless of the required k -anonymity. Besides the best coverage, t-MixDrones also provides a higher number of anonymizations per trip. It implies a higher entropy to the location privacy system, which can hamper the success of the adversary.

Furthermore, t-MixDrones keeps the same resilience of the MixDrones approach facing a trajectory-based linking attack. These approaches provide protection higher than 50% of the traditional MZ mechanism. Also, the drone's airway change performance causes an increasing travel time of less than 8% in all scenarios. Considering the drone's SWaP limitation and the average travel time, this delay has a minor impact on the network. However, the inherent characteristics of the provided application service must

be considered to evaluate the effects of this additional time properly.

In a nutshell, t-MixDrones is the LPPM to provide a better location privacy level to drones in the IoD environment. Overall, t-MixDrones overcomes both MixDrones and the traditional MZ mechanism, mainly providing a higher coverage and re-anonymization rate. Therefore, t-MixDrones can be applied in real-world IoD environments with dense traffic of drones, being an enhanced and lightweight security protocol to ensure the drone's location privacy.

Also, this evaluation highlights new challenges to be further investigated:

- A major shortcoming faced by t-MixDrones is the potential energy consumption increase caused by the additional maneuvers from the altitude change. Indeed, even with a small *ITTR*, it represents a small parcel of flight time which intuitively means a small growth of energy consumption.
- The provided QoS must be investigated thoroughly, mainly regarding the *ITTR* as a key “side effect” of the drone's airway change. This mechanism goal is currently performed considering two factors: the probability of changing the airway and the availability of intermediary airways. Hence, QoS factors can also be considered, such as the delay tolerance;
- Some configuration parameters of t-MixDrones should be tuned dynamically according to the environmental topology. For instance, the MZ coverage radius must adapt to a length that provides a proper coverage and anonymization rate;

4.3 MixRide

In the last section, we present the t-MixDrones, the first LPPM designed specifically for IoD. Although t-MixDrones presented a suitable performance regarding location privacy, the mechanism is not an energy-aware approach. It means that its utilization can increase the drone power consumption, affecting the flight time and, therefore, the QoS. Therefore, new strategies must be designed to overcome this shortcoming.

In recent years, the cooperation between aerial and grounded transportation has arisen as a promising approach to enhancing the drone's energy efficiency [128, 129, 130]. In these approaches, drones take a ride on ground transportation to save or recharge their batteries, for instance, on the bus roof. These places can represent a silent zone since they can accommodate many drones. Hence, the design of novel LPPMs can explore this characteristic.

Bearing this cooperation in mind, we design MixRide, an energy-aware LPPM for the IoD. In a nutshell, MixRide provides location privacy through aerial-grounded vehicle collaboration, where drones take a ride with grounded vehicles. MixRide assigns a vehicle for the drone to land and remain in silent mode. Hence, the ground vehicle acts as a mobile MZ, where the drones change their pseudonyms while saving energy.

As occurred in the t-MixDrones proposal, we follow the framework guidelines to design MixRide. In the next subsections, we: describe the application scenario and threat model; present existing approaches regarding an air-to-ground collaboration; model MixRide formally; conduct a performance evaluation to compare our solution with t-MixDrones; and discuss the impact of rides on the drone QoS in terms of delay, battery power consumption, communication channel usage, and the level of location privacy.

4.3.1 Threat Model

Both the application scenario and threat model are similar to the ones described for t-MixDrones (Section 4.2). In general, the referred IoD scenario embraces an urban environment with a high density of drones, and buses are part of a ground public transportation system. Furthermore, We consider IoD as a transportation environment where the available airways are parallel to the terrestrial roads at different altitudes. Also, we assume that the communication channel between the nodes is entirely reliable.

As MixRide provides location privacy through successive re-pseudonymization when the drones land on the same vehicle, a trajectory-based de-anonymization attack can hamper this provided privacy. Therefore, we consider the adversary model described in Section 4.2.1, which aims to link the trajectories and then de-anonymize the generated pseudonyms. This attacker acquires a background knowledge of the network through the invasion of some unreliable Location Server in the network, including the anonymized mobility traces. The adversary assumes that drones fly following the shortest path given the source and destination points. Thus, he estimates this aspect considering all combinations of the sub-trajectories before and after the re-pseudonymization to address the shortest one, linking a former and a new pseudonym as belonging to a unique drone.

4.3.2 Related Studies

The collaboration between aerial and terrestrial transportation emerges as a proper strategy to mitigate the challenges posed by drone SWaP limitations. This approach entails the aerial vehicles hitching a ride on terrestrial ones, based on specific criteria, enabling the former to undertake their flight continuously. As the propellers consume a significant portion of the drone’s battery, the rides contribute to the energy efficiency.

Over the years, different studies explored this collaboration in different applications, as presented in Table 4.5. They are related with parcel delivery systems [129, 130, 131] and surveillance applications [128, 132]. Besides the ride process, some approaches investigated the viability of recharging the drones with devices attached to the ground vehicles [128, 130] or also to swap the battery [129].

On one hand, the majority of proposed scheduling approaches are based on Mixed Integer Linear Programming (MILP) [128, 132, 133] or even the application of suboptimal algorithms [129, 131]. In general, the scheduling process occurs “offline”, in other words, the system assigns a ride before the drone starts to fly. These approaches can provide a local-optimum result considering each specific study’s goal but they do not consider an important aspect of real-world transportation: the inherent delay caused by traffic jams. Therefore, these strategies do not cover realistic scenarios. On the other hand, some studies applied greedy-based heuristics focusing on assigning the rides “in-flight” [130]. With these strategies, the network system assigns the ride in an opportunistic way, as much as possible. Although it can mitigate the related ground transportation delay, the associated QoS can be affected since there is no “broader view” over the whole network. Therefore, new strategies must be designed to merge the best characteristics of each kind of approach.

Given this perspective, we design MixRide to overcome these issues, focusing on enhancing the drone’s energy efficiency while providing adequate location privacy. We address these aspects through the collaboration between the drones and the grounded transportation system. The grounded vehicle acts as a mobile silent MZ where the drones can change their pseudonym.

4.3.3 Design of MixRide

MixRide follows the concept of the MZ mechanism and, therefore, the k -anonymity principle. In this approach, the drone has a PoI to save its battery, taking place on a

Table 4.5: Related Studies of Aerial and Terrestrial Transportation Collaboration

Ref.	Colab.	Application	Goal	Scheduling Approach
[128]	Public buses	Video surveillance	Recharging the UAVs battery on the buses' roof, optimizing the video surveillance system	Mixed Integer Linear Programming (MILP) with a heuristic
[133]	Public buses	General coverage mission	Save the UAVs battery through scheduled rides on the buses' roof	MILP
[129]	Public train	Delivery	Swapping of the UAVs battery via an automatic battery swap system	Suboptimal task allocation algorithm
[132]	Public transportation vehicles	Surveillance	Save the UAVs battery aiming to reach far away surveillance areas	MILP with two suboptimal algorithms
[130]	Public buses	Last-mile delivery	Recharging the UAVs battery on the buses' roof	Greedy-based heuristic
[131]	General grounded vehicles	Delivery	Design of an incentive mechanism to promote the collaboration between grounded vehicles and UAVs, determining a ride's pricing process	Polynomial near-optimal algorithm
Ours	Public buses	General	Enhance location privacy through anonymizations while the drones save battery	Greedy-based heuristic

grounded vehicle (e.g., a bus roof). This PoI represents a mobile MZ that allows the drones to change their pseudonyms while saving battery.

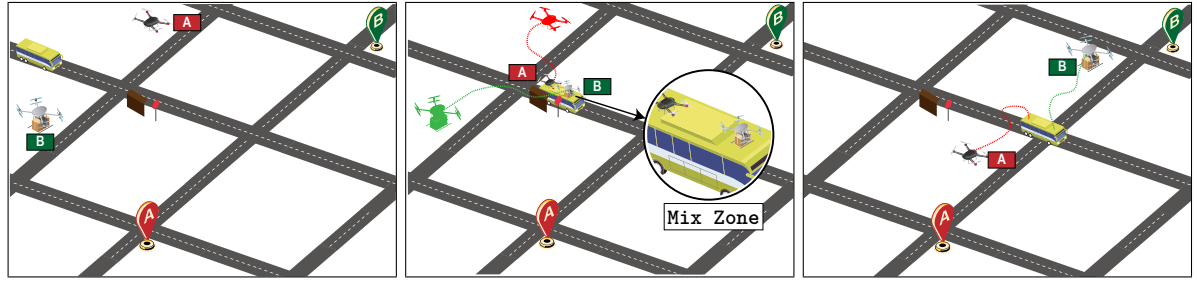
Fig. 4.10 illustrates the MixRide concept. In this scenario, two drones (A and B) fly over the environment, having distinct destinations. In Fig. 4.10a, a bus is near these drones, and it is approaching a bus stop. In this case, the IoD system identifies that both drones can take a ride on the bus to save energy. When the drones land on the bus' roof, the bus becomes a mobile MZ, where the drones can change their pseudonyms and remain in a silent mode (Fig. 4.10b). When the bus passes through a given checkpoint, the drones leave the roof and fly to their respective destinations (Fig. 4.10c).

In terms of the drones' mobility traces, when they are on the roof of the bus, they do not update their location as they are in the silent zone (MZ). So, when they leave the MZ, the traces are associated with new pseudonyms, which can increase the network's location privacy.

► **Formal definition:** To support the MixRide formal definition, we assume the following system model:

- \mathcal{D} is the set of available drones in the IoD system. Each drone $d \in \mathcal{D}$ has the following attributes. ε : the current energy level. $PP = \langle p_0, \dots, p_{|PP|} \rangle$: the current path planning, composed by $|PP|$ geographic checkpoints. This path planning considers

Figure 4.10: MixRide concept



(a) Two drones fly near to a bus
 (b) When the bus stops at a station, they take a ride on the bus' roof, which acts as a MZ
 (c) When the drones are close to their destination, they leave the bus' roof

Source: Elaborated by the author

only the aerial space, which means that it does not embrace a potential ride; l : the current location, denoted by a given geographical coordinated system;

- \mathcal{GV} is the set of available vehicles in the grounded system. Each vehicle $v \in \mathcal{GV}$ has the following attributes. max_d : the maximum number of drones that the vehicle can carry; $S = \langle s_0, \dots, s_{|S|} \rangle$: the ordered sequence of stops associated with its current trip; l : the current location; $C = \langle c_0, \dots, c_{|S|-1} \rangle$: the ordered sequence of allocated drones between two consecutive stops. Thus, the i^{th} element is related to the trajectory between the stops $s_i \rightarrow s_{i+1}$;
- $Trip_d^v = \langle PP^b, S^r, PP^f \rangle$ represents a collaboration trip between a drone $d \in D$ and a grounded vehicle $v \in \mathcal{GV}$. It comprises three sub-sequences from the drone's path planning $d.PP$ and the grounded vehicle's stops $v.S$. PP^b is a sub-sequence of $d.PP$ represents the initial aerial trip until the drone starts the ride; S^r is a sub-sequence of $v.S$ and represents the geographical points during the grounded ride; PP^f is also a sub-sequence of $d.PP$, but represents the final part of the aerial trip when the drone leaves the ride and flies until the destination point.

The aerial-grounded collaboration has a trade-off: while the ride can save the drone's battery power, it also causes a delay in the drone's travel since the grounded vehicle's average speed is lower than the drone's speed [129, 130]. Thus, we can define three different environmental factors that affect the MixRide performance: the k -anonymity level, the ride's delay, and the battery's power consumption. Considering the k -anonymity principle, the higher the k value, the higher the system's entropy, leading to a higher location privacy.

Therefore, we adopt an opportunistic approach, in which MixRide assigns drones to take a ride as much as possible, following the max_d restriction. Once the MixRide allocates a given drone d to take a ride on a grounded vehicle v , the initial point of the ride $Trip_d^v$ is already defined, but it is necessary to determine where the ride will finish.

Algorithm 6: MixRide

```

Input   :  $d, \tau, w_0, w_1$ 
Output :  $Trip_d^v$ 
1  $\mathcal{G}\mathcal{V}_{cand} \leftarrow \emptyset, Trip_d^v \leftarrow \emptyset, ride_{best} \leftarrow \infty$ 
2 foreach  $v \in V$  do
3    $v_{stp} \leftarrow$  index of  $v.S$  element related to the next  $v$ 's stop
4   if  $distance(d.L, V.S_{stp}) \leq \tau \wedge v.C_{v_{stp}} < v.max_d$  then
5      $\mathcal{G}\mathcal{V}_{cand} \leftarrow V_{cand} \cup v$ 
6 foreach  $v \in V_{cand}$  do
7    $v_{stp} \leftarrow$  index of  $v.S$  element related to the next  $v$ 's stop
8    $PP^b \leftarrow \langle d.PP_0, \dots, nearest(d.PP, v.S_{v_{stp}}) \rangle$ 
9    $S_{cand} \leftarrow \langle v.S_{v_{stp}+1}, \dots, v.S_{|v.S|} \rangle$ 
10  foreach index  $s$  of  $S_{cand}$  do
11    if  $v.C_s < v.max_d$  then
12       $S^r \leftarrow \langle v.S_{v_{stp}}, \dots, v.S_s \rangle$ 
13       $PP^f \leftarrow \langle nearest(d.PP, v.S_s), \dots, d.PP_{|d.PP|} \rangle$ 
14       $Trip' \leftarrow \langle PP^b, S^r, PP^f \rangle$ 
15       $ride \leftarrow w_0 \mathcal{T}_{rate}(d, Trip') + w_1 \mathcal{E}_{rate}(d, Trip')$ 
16      if  $ride < ride_{best}$  then
17         $ride_{best} \leftarrow ride$ 
18         $Trip_d^v \leftarrow Trip'$ 
19    else
20      goto Line 6

```

This endpoint is the fundamental factor in minimizing the impact of the ride's delay and the battery's power consumption.

It leads us to the formal definition of MixRide (Algorithm 6). The algorithm is a ZSP-centered method, called in every drone's location update. In this approach, we assume that a given ZSP $z \in \mathcal{Z}$ is aware of the path plannings of all drones and grounded vehicles under its geographic region. As input, the algorithm requires the data about the drone d , a given distance threshold τ , and two weight values w_0 and w_1 related to the impact of the ride's delay and the battery consumption, respectively. As output, the algorithm generates the collaborative trip $Trip_d^v$.

Besides the output, the algorithm considers a set $\mathcal{G}\mathcal{V}_{cand} \subset \mathcal{G}\mathcal{V}$ that stores the grounded vehicles that can provide a ride to d , and a variable $ride_{best} \mapsto \mathbb{R}^+$ that stores the value of the best ride, based on a minimization function, which is described further. Before calculating the ride's cost, it is necessary to address which grounded vehicles can provide a ride. It occurs by evaluating of the distance between the drone's current location $d.L$ and the next stop location of a given vehicle $v \in \mathcal{G}\mathcal{V}$, and the availability of a vehicle "slot". If both conditions are satisfied, v is added as a candidate (Lines 2-5).

After, MixRide evaluates each candidate, considering all the possible rides (Lines 6-20). It is modeled step-by-step. Firstly, PP^b is taken from the source geographic point

that d started its trip to the aerial geographic point nearest to the next v 's stop (Line 8). The next step consists of defining S^r , which effectively represents the ride. As the initial geographic point $v.S_{v_{stp}}$ is already defined, the candidates' endpoint must range from the next immediate point to the last one. They compose the sequence S_{cand} (Line 9). Each endpoint candidate s of S_{cand} can pose a different configuration regarding the ride's delay and d 's battery consumption. Hence, they need to be assessed one by one (Lines 10-18). The first evaluated aspect is the current number of allocated drones until the endpoint s . If there is no available spot, s is not a valid endpoint. As the ride must follow the ordered stops $v.S$, all the following endpoints cannot be considered since v will be "full" at the stop $v.S_s$ (Lines 19-20). Otherwise, it is possible to offer a ride with $v.S_s$ as the endpoint. Thus, the algorithm defines a ride S^r from the next v 's stop $v.S_{v_{stp}}$ to $v.S_s$ (Line 12). PP^f is taken from the aerial geographic point nearest the ride's endpoint to the actual drone's destination (Line 13). These three sub-sequences (PP^b , S^r , and PP^f) compose a candidate trip $Trip'$.

Hence, the ride can finally be evaluated for its potential delay and energy consumption through two distinct functions, described in Equations 4.11 and 4.12, respectively (Line 15). \mathcal{T}_{rate} is given by the rate of $d.PP$ and $Trip_d^v$, considering their predicted time interval to reach the destination point. Thus, MixRide aims to provide a lower \mathcal{T}_{rate} . The \mathcal{E}_{rate} gives the battery's power consumption as a predicted energy power consumption rate that d has to perform the $Trip_d^v$ and $d.PP$, respectively. Also, MixRide intends to provide a lower \mathcal{E}_{rate} , which indicates that the ride allows the drone to save a significant amount of power. The results of these functions are tuned through the input weights w_0 and w_1 , and the resulting sum represents the ride score. If the *ride* score is less than the current best $ride_{best}$, the trip candidate $Trip'$ becomes the new best ride (Lines 16-19). After the algorithm evaluates all the candidates and potential rides, $Trip_d^v$ represents the best ride configuration. Therefore, the related ZSP sends this data to drone d , which will replace its original path planning $d.PP$. The drone's approximation, landing, and takeoff occur synchronously with the grounded vehicle. These maneuvers are not limited nor dependent on MixRide and can be applied according to the well-known methods of the literature [129, 130].

$$\mathcal{T}_{rate}(d, Trip_d^v) = 1 - \frac{\text{predTime}(d.PP)}{\text{predTime}(Trip_d^v)} \quad (4.11)$$

$$\mathcal{E}_{rate}(d, Trip_d^v) = \frac{\text{predEnrgConsump}(d.E, Trip_d^v)}{\text{predEnrgConsump}(d.E, d.PP)} \quad (4.12)$$

4.3.4 Simulation Setup and Performance Evaluation

Considering the obligatory presence of ground vehicles, we consider an urban environment where buses are part of a public transportation system. We use the RioBuses dataset [134] to support our experiments. RioBuses has GPS data gathered from the public transportation vehicles in Rio de Janeiro, Brazil. We delimit the experiments over Ipanema Beach, where buses have a large concentration. The region contains around 15 km of roads.

As in the t-MixDrones setup, we use the IoDSim as a simulation tool of the IoD environment. We spread 50 drones over that scenario, reaching a maximum speed of 15 m/s. The energy consumption model follows the drone MD4-300, commonly used in delivering goods, whose expected time flight is 45 minutes [135]. Thus, the drone does not recharge its battery during the simulation. There are 24 buses available to give a ride, carrying a maximum of 3 drones. Each bus moves with a maximum speed of 12 m/s through distinct routes, based on the Riobuses dataset [134].

Regarding the communication model, IoD nodes have the same protocols and radio configurations, using TCP, AODV, and CSMA/CA. The radio operates following the standard 802.11n with modulation mode of 1×1 20 MHz, having a maximum throughput of 72.2 Mbps. Regarding the MixRide parameterization, we consider a distance threshold τ of 50 m, which delimits a ride-matching between nearest drones and buses. To evaluate the impact of the weights w_0 and w_1 over the mechanism, we define three configurations:

- C1 ($w_0 = 0.25, w_1 = 0.75$): it prioritizes rides with less predicted delays;
- C2 ($w_0 = 0.5, w_1 = 0.5$): it prioritizes rides that balance both the delay and power consumption;
- C3 ($w_0 = 0.75, w_1 = 0.25$): it prioritizes rides with less predicted power consumption;

► **Metrics:** Based on the discussed goals, we consider four metrics, such that one of them is *TMA*, already presented in Section 4.2.7. The remained metrics are described as follows.

- *Trip Delay Rate (TDR)*: based on the duration time Δt_a of a baseline trip (aerial only), the TDR calculates the delay rate Δt_m of the same trip when influenced by an LPPM, such as MixDrones and MixRide. It is defined as follows.

$$TDR = \frac{\Delta t_m}{\Delta t_a} \quad (4.13)$$

Table 4.6: Simulation Parameters regarding the MixRide performance evaluation

Parameter	Value
Simulation time	30 minutes
#Drones	50
Drone max. speed	15 m/s
Drone's Energy Model	MD4-300 [135]
#Buses	24
Bus max. speed	12 m/s
\max_d	3
τ	50 m
Configuration of weights w_0, w_1	{C1, C2, C3}

- *Power Consumption Rate (PCR)*: Similar to *TDR*, it measures the rate between the battery's power consumption ε_a and ε_m , where they represent the consumption during a trip through the air and when an LPPM influences it, respectively. It is formally defined as follows.

$$PCR = \frac{\varepsilon_m}{\varepsilon_a} \quad (4.14)$$

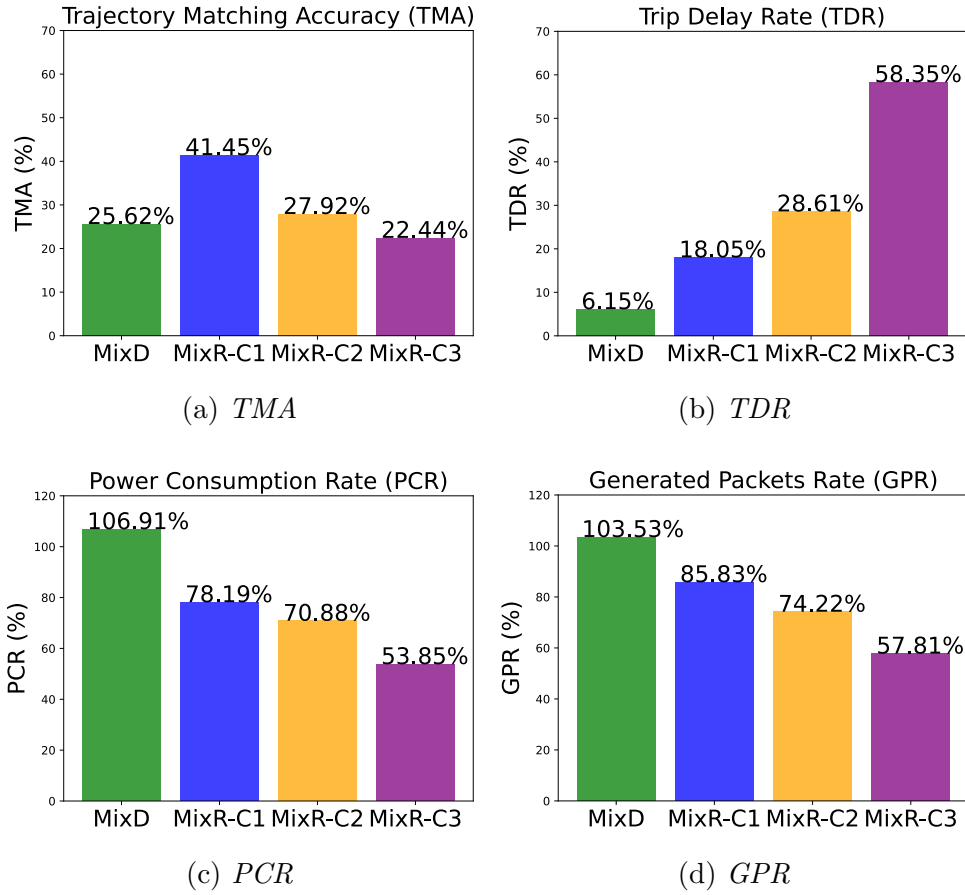
- *Generated Packets Rate (GPR)*: this metric calculates the rate between the number of generated packets of the baseline trip Pkt_a and its corresponding when applying a mechanism Pkt_m . With this metric, we can analyze what mechanism consumes the most of the communication channel. It is given by:

$$GPR = \frac{Pkt_m}{Pkt_a} \quad (4.15)$$

4.3.5 Results and Discussion

This section presents the results of the experimental evaluation. They were obtained through the average of 35 replications for each evaluated mechanism and configuration with 95% of interval confidence. To perform a fair comparison, we simulate the replications in an environment without utilizing of any LPPM, named the "baseline experiments". For all results, the interval errors are less than 1%. Thus, they are not represented in the charts. Also, we define the following acronyms: MixD represents the t-MixDrones mechanism; MixR-C1, MixR-C2, and MixR-C3 represent the MixRide mechanism, parameterized with the configurations above.

Figure 4.11: Results of the MixRide performance evaluation



Source: Elaborated by the author

► **Trajectory Matching Accuracy (*TMA*)**: Fig. 4.11a shows the results of *TMA* for the evaluated LPPMs. As we can note, MixRide with configurations that do not prioritize the related delay (C2 and C3) provides a privacy level similar to the MixDrones. Indeed, configuration C3 provides higher location privacy than t-MixDrones, where less than 25% of the trajectories are de-anonymized. The balanced configuration (C2) also addresses a proper privacy level, with a *TMA* similar to the t-MixDrones. It indicates that C2 can be used in services in which drones have reasonable energy efficiency and do not need to perform fast flights.

On the other hand, when MixRide prioritizes the rides with less predicted delay (C1), the attacker can de-anonymize about 42% of the trajectories. This high value is associated with the short time interval that a given drone remains on its ride, in which there is no time for another drone to enter the MZ of the same vehicle. Thus, there are situations in which the drone remains alone until it leaves the vehicle, which facilitates the attack's success since there are no other trajectories to link.

► **Trip Delay Rate (*TDR*):** For *TDR*, we consider only the trips a given drone takes a ride. This selection makes the comparative evaluation unbiased. Fig. 4.11b presents the *TDR* results. Following the premises of each configuration, the lower the weight w_0 , the lower the delay. Although the delay increases in the balanced configuration (C2), it is less than 30% of the baseline trip, on average. It highlights that balanced weight configurations can be a proper choice for delay-tolerant services. The rides following the configuration C3, in turn, present an average delay higher than 50%, which can indicate a loss of QoS for services that demand a fast performance.

Also, we can note that t-MixDrones presents a delay lower than all configurations of MixRide. However, the t-MixDrones-related trajectory is similar to the baseline one. The difference consists in changes of the airways resulting from the application of the mechanism. However, all the flight is performed through the air, demanding more battery power, as described in the following result.

► **Power Consumption Rate (*PCR*):** As in *TDR*, we also consider only the trips in which the drone takes a ride. The *PCR* results, presented in Fig. 4.11c, highlight an opposite scenario to *TDR*. t-MixDrones presents a rate higher than 100%, indicating that the trips consume more battery power than the baseline ones, on average. This result occurs due to the airway change demanded by the mechanism, expending an additional effort on the drone's propellers and, therefore, its battery power. Regarding the MixRide, all the configurations can save the power consumption compared to the baseline trip. As occurs with the *TDR* metric, the results point out that the premises of each configuration are properly followed. Thus, the lower the weight w_1 , the lower the *PCR*. Specifically, almost half of power consumption is saved in configuration C3. Considering the *TDR* of C3, this economy occurs due to the longer rides assigned to the drones, where they remain in the silent mode for a longer time.

Configurations C1 and C2, in turn, consume a power rate of about 78% and 71% of the baseline trip, on average. Analyzing these results with the *TDR*, we note that their performance has a similar interval difference. However, when we consider the evaluated *TMA*, C2 highlights a better configuration in terms of location privacy.

► **Generated Packets Rate (*GPR*):** Once again, to keep the evaluation unbiased, we only consider the packets generated by the drones. Fig. 4.11d shows the results of *GPR*. These rates are similar to *PCR* because the generation of communication packets in the MixRide is related to the time interval that the drones remain in a silent zone and, therefore, save power.

Furthermore, we can note that configurations that prioritize the assignment of rides with less power consumption also avoid a potential communication channel congestion. On the other hand, t-MixDrones generates more packets than the baseline scenario. This

issue occurs because the drones remain inside the MZ over a small interval. Moreover, the number of packets related exclusively to the mechanism overlaps the number of non-transmitted packets when the drone stays in the MZ.

► **General Discussion:** This experimental evaluation reveals that MixRide can provide location privacy to the IoD at the same level as t-MixDrones while improving the drone’s power consumption and reducing the number of generated data packets. However, the MixRide configuration affects these improvements deeply.

Configurations that prioritize the assignment of rides with less power consumption can provide an adequate level of location privacy, reducing the number of generated packets, besides the battery’s power economy. However, the drone’s trip has an increased delay, representing an issue when the provided service does not tolerate significant delays.

On the other hand, configurations that prioritize rides with a lower delay can allow the drones to assign rides with delay rates lower than 20% while saving a parcel of battery. However, the provided location privacy is low compared to the other configurations. In this case, MixRide can be integrated with other LPPMs (e.g., t-MixDrones), in which the IoD management can learn through the environment the best way to apply each one in a smart approach.

Summarily, the results point out the following challenges to be further investigated:

- It is fundamental to carry out a comprehensive study regarding the dynamic tuning of MixRide configuration in such a way the mechanism can adapt to the environmental conditions and the provided service requirements. AI-based techniques can be applied to mitigate this shortcoming, for instance, the designing of RL-based strategies;
- Given a real-world service performed by the drone, the analysis of the required QoS is a paramount aspect of configuring MixRide accordingly. For instance, delay-tolerant services can use the MixRide with a configuration that tries to reduce power consumption. Whether the environment has several recharge stations, balanced configurations can be a proper choice, providing adequate location privacy;
- MixRide is susceptible to an absence of available rides since the mechanism considers a heuristic approach to assign them, based on a “regional exploration” of available rides. Hence, enhanced ride-scheduling strategies shall be designed to optimize the location privacy performance, the ride-related bus load, and the trade-off regarding power consumption and delay.

4.4 TDG

In the former sections, we presented two MZ-based LPPMs. However, they were designed to be deployed in dense scenarios, only. Hence, there remains a lack of LPPMs for sparse environments in the IoD. Although DaaS technologies are currently growing, some services will still be provided by a few drones, for instance, restricted surveillance or mobile wireless coverage [1]. As discussed in Section 3.2.2, dummy-based protocols are a group of LPPMs that can provide location privacy in traditional mobile networks for sparse scenarios [83]. It aims to protect a node’s location through spatial cloaking, generating dummy positions in a way that they compose an indistinguishable set, simulating a group with real and fake nodes. Hence, the design of this mechanism for IoD can overcome this current lack.

We propose the Topology-based Dummy Generation (TDG) LPPM for the IoD paradigm. This mechanism focuses on the IoD topology characteristics regardless of the presence of nearest drones, meeting the requirements to be applied in a sparse scenario. To the best of our knowledge, TDG is the first approach for sparse scenarios in IoD, contributing to the advancement of the state of the art in this field.

In this section, following the framework to design LPPMs, we present a threat model based on an IoD spatiotemporal inference attack (IoD-STIA). To better understand the current literature, we analyze the recent studies regarding dummy-based strategies. From that, we design TDG, discussing its key characteristics, how it can be applied in the IoD environment. We also evaluate the TDG performance regarding the communication channel usage and facing IoD-STIA through a series of simulations.

4.4.1 Threat Model

The drone’s trajectory planning is an optimization problem that considers several requirements to compute the best path [136], where the airspace is modeled as “free to flight”. IoD, in turn, has a traffic topology composed of well-defined airways, constraining the flyable airspace. Thus, it is necessary to plan the drone’s trajectory into the airways’ boundaries. Thus, it is reasonable to formulate the hypothesis that in the IoD environment, the planning of a drone’s trajectory aims to optimize a set of requirements, preserving the air traffic topology. Inductively, if we divide a trajectory \mathcal{T} in n sub-trajectories τ_0, \dots, τ_n , all sub trajectory τ follows this hypothesis. In other words, a node’s trajectory is optimal if all the sub-trajectories are also optimal.

Based on this hypothesis, we design an IoD-based Spatio-temporal Inference Attack (IoD-STIA) that selects a group of queries \mathcal{Q}' through the processing of a trajectory $\mathcal{T}_{\mathcal{A}}$ in which its spatio-temporal coordinates must correspond to the coordinates of the real queries $\mathcal{Q}_r \subset \mathcal{Q}$. In other words, the attacker tries to infer the actual drone positions assuming that they represent the optimal trajectory among all the possibilities. Algorithm 7 describes IoD-STIA. As input, it requires the topological graph G , the generated system's queries \mathcal{Q} , an integer threshold n that limits the number of generated trajectories in each iteration, and an optimization function f . In this study, we consider as f the standard deviation of the drone's average speed along the sub-trajectories. however, other functions can be considered, and our proposal is not limited nor dependent on this function.

Algorithm 7: IoD-STIA

Input : G, \mathcal{Q}, f, n
Output: \mathcal{Q}'

- 1 $\mathcal{T}_{heap} \leftarrow$ initialize an empty heap
- 2 $\mathcal{Q}_{clusters} \leftarrow$ process ordered clusters of \mathcal{Q} based on $\mathcal{Q}.t$
- 3 **for** $cluster \in \mathcal{Q}_{clusters}$ **do**
- 4 **if** $|\mathcal{T}_{heap}| = 0$ **then**
- 5 **for** $query \in cluster$ **do**
- 6 $\mathcal{T}_{heap}.insert(\langle query \rangle, f(query))$
- 7 **else**
- 8 $\mathcal{T}'_{heap} \leftarrow$ initialize an empty heap
- 9 **for** $\tau \in \mathcal{T}_{heap}$ **do**
- 10 **for** $query \in cluster$ **do**
- 11 $\tau' \leftarrow \langle \tau, query \rangle$
- 12 $\mathcal{T}'_{heap}.insert(\tau', f(\tau'))$
- 13 $\mathcal{T}_{heap} \leftarrow n$ first elements of \mathcal{T}'_{heap}
- 14 $\mathcal{Q}' \leftarrow$ the first element of \mathcal{T}_{heap}

The attack works as follows. Firstly, we initialize an empty heap of trajectories \mathcal{T}_{heap} that stores the generated trajectories (Line 1). After, the queries are grouped in a sequence of clusters $\mathcal{Q}_{clusters}$, ordered by the query's timestamp t (Line 2). Each cluster will contribute to a single coordinate in the optimal trajectory. Hence, it is necessary to process each sub-trajectory candidate along with these clusters (Lines 3–13).

At the beginning of the first iteration, there are no generated sub-trajectories. Thus, each query that belongs to the first cluster is taken as the first trajectory point, ordering them by f (Lines 4–6). For all the other clusters (Lines 7–13), the sub trajectories from \mathcal{T}_{heap} are combined with each query of the current cluster and inserted in an auxiliary heap \mathcal{T}'_{heap} , also following the optimization function f (Lines 9–12). As these combinations represent a geometric progression, only the n best trajectories are kept along

Table 4.7: Recent studies related to dummy-based LPPMs

Ref.	Network	Generation Method	Architecture
[137]	Cellular	- Caching-aware dummy selection	Client-side
[139]	General	- Probabilistic generative model	Client-side
[138]	Cellular	- Spatiotemporal correlation filtering	Client-side
[82]	VANETs	- Circle-based region	TTP
[83]	Cellular	- Decoy queries	Client-side
[84]	General	- Plausible dummies	Client-side
[85]	General	- Attribute-aware dummies	TTP
Ours	IoD	- Topology-based	TTP

the process (Line 13). After all the cluster processing, the first element of \mathcal{T}_{heap} has the optimal trajectory coordinates, representing the inference carried by the IoD-STIA.

4.4.2 Related Studies

Over the years, several dummy-based methods have been proposed, considering their environmental characteristics and constraints. In Cellular Networks, the strategies explored human user behavior. The dummies were generated considering regions with similar query frequency by users, addressing a query caching [137]; a filtering technique exploring time reachability and direction similarity of users' movements [138]; and the imitation of real user mobility patterns being generated at the same time as the actual query [83]. Due to the absence of a trusted entity in Cellular networks, all the studies used the Client-side architecture, where the node generates the dummies by itself and communicates with the LBS directly. The TTP architecture has been used when the environment has constraints regarding energy, communication, or computational cost. In VANETs, for instance, Arif et al. [82] proposed a circle-based dummy generation allied with an anonymizer as TTP, transferring this process from the cars to a trusted party.

Some studies investigated the dummy-generation methods being applied to mobile networks in general. Hence, these methods are based on common characteristics of mobile behavior. For instance, generating the dummy queries through a probabilistic mobility model using a real trace dataset [139]; considering an optimal mobility trajectory over the time [84]; and exploring the variability of query attributes rather than the localization, only [85].

These related studies pointed out that dummy-based strategies have been adapted over the years according to the environmental characteristics and nodes' mobility patterns, always aiming to be resilient in facing different location-based attacks. Nonetheless, there

is no study that investigates this mechanism in the IoD paradigm. Considering that the mechanism will generate many queries and send them through a communication channel, it can lead to a high traffic flow.

Although some proposed generation strategies can easily fit in IoD, this issue is a barrier. The high mobility dynamics and the available airspace are also constraints to generating concise dummy queries. As previously discussed, a drone is able to fly over well-defined airways with boundaries, direction, and other features, such as speed limit. Besides the airways, the drone can freely fly over specific PoIs, expressed through the nodes. These factors pose a complex mobility behavior, leveraging the design of enhanced LPPMs that need to handle these challenges.

4.4.3 Design of TDG

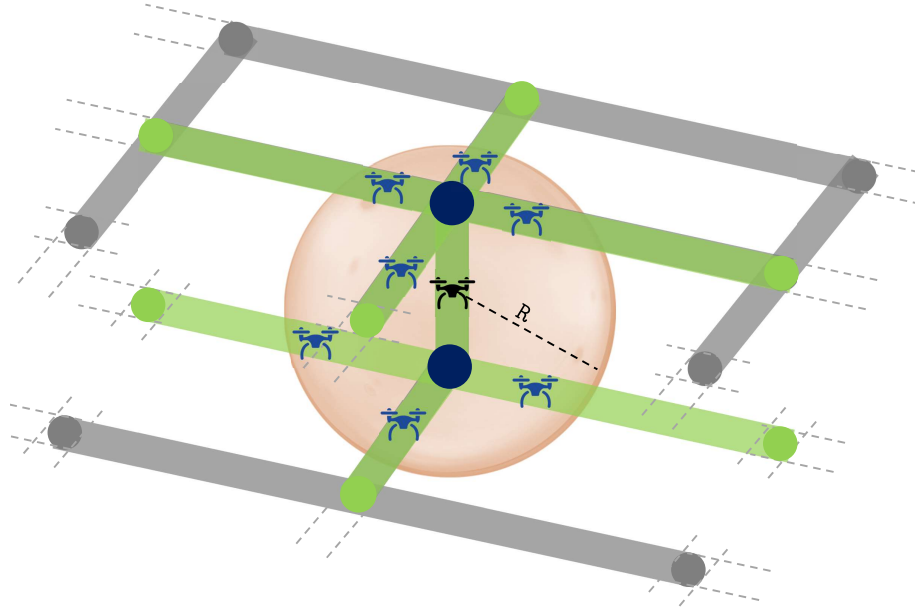
The Topology-based Dummy Generation (TDG), as the name suggests, is mainly based on the IoD topology to generate dummy queries, regardless of the presence of the nearest drones. In this approach, we combine different concepts from well-established mechanisms: the neighborhood random movements and the generation over circle-based regions [82].

Figure 4.12 illustrates the TDG concept. A current drone (black drone) will always have previous and subsequent topological points n_{prev} and n_{subs} , respectively (blue circles). Thus, the dummy query positions (blue drones) can be generated over their adjacent airways segments (green segments). However, in sparse scenarios, these adjacent points can be far away from each other, easing inference attacks since the dummies can be placed sparsely [140]. To handle this issue, we consider a spherical region with a radius r whose goal is to restrict the dummies' distance from the actual position.

Algorithm 8 formally defines TDG. Here, we assume that given a drone d , a ZSP \mathcal{Z} manages the d 's path planning ρ_d , providing information about the airways topology through a graph $G = (V, E)$. Furthermore, the algorithm requires as input the restriction radius r ; the required k -anonymity level k ; the drone's identifier id and location l_d ; the timestamp t ; and the required query information q_{info} .

Initially, the mechanism initializes the adjacent airways set E_{adj} as an empty set (Line 1) and assigns the real drone's query to the output queries set \mathcal{Q} (Line 2). After, it obtains the closer points n_{prev} and n_{subs} based on the current drone's location l_d and the topological graph G (Line 3). The next step consists of selecting the adjacent airways to place the dummies. Thus, the adjacent edges of both n_{prev} (Lines 4–5) and n_{subs} (Lines 6–7) are selected.

Figure 4.12: Topology-based dummy generation concept



Source: Elaborated by the author

Algorithm 8: TDG

Input : $G, R, k, id, l_d, t, q_{info}$
Output: Q

- 1 $E_{adj} \leftarrow \emptyset$
- 2 $Q \leftarrow \{\langle id, t, l_d, q_{info} \rangle\}$
- 3 $n_{prev}, n_{subs} \leftarrow \text{closerPoints}(G, l_d)$
- 4 **for** $v \in G.adjacents(n_{prev})$ **do**
- 5 $E_{adj} \leftarrow E_{adj} \cup \{\langle n_{prev}, v \rangle\}$
- 6 **for** $v \in G.adjacents(n_{subs})$ **do**
- 7 $E_{adj} \leftarrow E_{adj} \cup \{\langle n_{subs}, v \rangle\}$
- 8 **while** $|Q| \leq k$ **do**
- 9 $\langle n_1, n_2 \rangle \leftarrow$ choose a random tuple from E_{adj}
- 10 $distance \leftarrow \infty$
- 11 **while** $R < distance$ **do**
- 12 $l_{dummy} \leftarrow$ choose a random coordinate between the line segment of $\langle n_1, n_2 \rangle$
- 13 $distance \leftarrow \text{calcDistance}(l_d, l_{dummy})$
- 14 $Q \leftarrow Q \cup \{\langle id, t, l_{dummy}, q_{info} \rangle\}$

Finally, the mechanism generates the dummy queries until the k level is satisfied (Lines 8–14). By randomly choosing an adjacent airway (Line 9), the dummy location l_{dummy} is selected, meeting the radius constraint (Lines 11–13). Given that we can consider the airway as a line segment over a 3D space, TDG randomly selects l_{dummy} from this segment (Line 12), which will compose a new dummy query (Line 14).

► *Time Complexity Analysis*: TDG has a time complexity cost also related to IoD

topology, as we described as follows. Firstly, it is important to note that TDG has several set-based operations. We assume that the mechanism implements the well-known optimal elementary set operation algorithms, whose costs are $\Theta(1)$. It is necessary to check all edges E to calculate the closer points of a location l_d (Line 3), leading to a cost $\Theta(|E|)$. Moreover, the building of adjacent airways set S occurs through the iteration of adjacent sets. Given that G is a directed graph and implements these sets with linear lists, each iteration process has time complexity $\mathcal{O}(|E|)$. The dummy queries generation (Lines 8–14) iterates for k times, where its inner operations have $\Theta(1)$ cost. Thus, the iteration complexity is $\Theta(k)$. Hence, the time complexity $T(TDG)$ is given by Equation 4.16.

$$\begin{aligned} T(TDG) &= 2\Theta(1) + \Theta(|E|) + 2\mathcal{O}(|E|) + \Theta(k) \\ &= \mathcal{O}(\max\{|E|, k\}) \end{aligned} \tag{4.16}$$

4.4.4 Simulation Setup and Performance Evaluation

With TDG, we focus on analyzing the environments where the applications are strictly related to the drone’s location privacy. For instance, an application where a drone can monitor the grounded traffic flow, requiring periodically contextual information to an LBS, demanding the drone’s location. Hence, the dummy-based mechanisms must be aware of the drone’s flyable airspace rather than third-party users. Hence, in the performance evaluation, we consider an urban scenario where the drones have well-defined path plannings managed by the ZSPs. They communicate with an LBS whenever there is a location request involved. The communication architecture used with the LBS is independent of the drone application module, which is a decision of the network infrastructure.

We perform simulations using the IoDSim. Table 4.8 shows a list of relevant simulation parameters. We design an urban environment that reflects a region of Manhattan, NY, considering a single airspace zone where the flyable airspace follows the grounded roads in two different altitudes. Since our mechanism focuses on sparse scenarios, we place fifteen drones with different path plannings over the environment. They move with a speed varying from 5 to 10 m/s and a Gauss-Markov mobility pattern.

Regarding the communication model, IoD nodes have the same protocols and radio configurations, using TCP, AODV, and CSMA/CA. The radio operates following the standard 802.11n with modulation mode of 1×1 20 MHz, having a maximum throughput of 72.2 Mbps. Besides the communication model, we vary both the k -anonymity level and the TDG radius to evaluate how these factors impact location privacy. For each

Table 4.8: Simulation parameters regarding the TDG performance evaluation

Parameter	Value
<i>k</i> -anonymity	{4, 5, 6, 7, 8, 9, 10}
TDG radius	{50, 100, 150, 200, 250}
Communication model	{Client-server, TTP}
#Drones	15
Drone speed	5–10 m/s
Mobility pattern	Gauss-Markov
Environment boundaries	900 × 1100 × 200 m ³
#Airways	2 (100 m and 150 m)
ZSPs	1
Simulation time	10 minutes

combination, we conducted 35 experiments considering 35 different simulation seeds to obtain statistical validation and variation with 95% of a confidence interval, totaling 2450 distinct experiments.

► **Metrics:** We consider four different metrics to evaluate the proposed solution. The first one brings a communication channel analysis. The other three evaluate the provided location privacy through the IoD-STIA point of view.

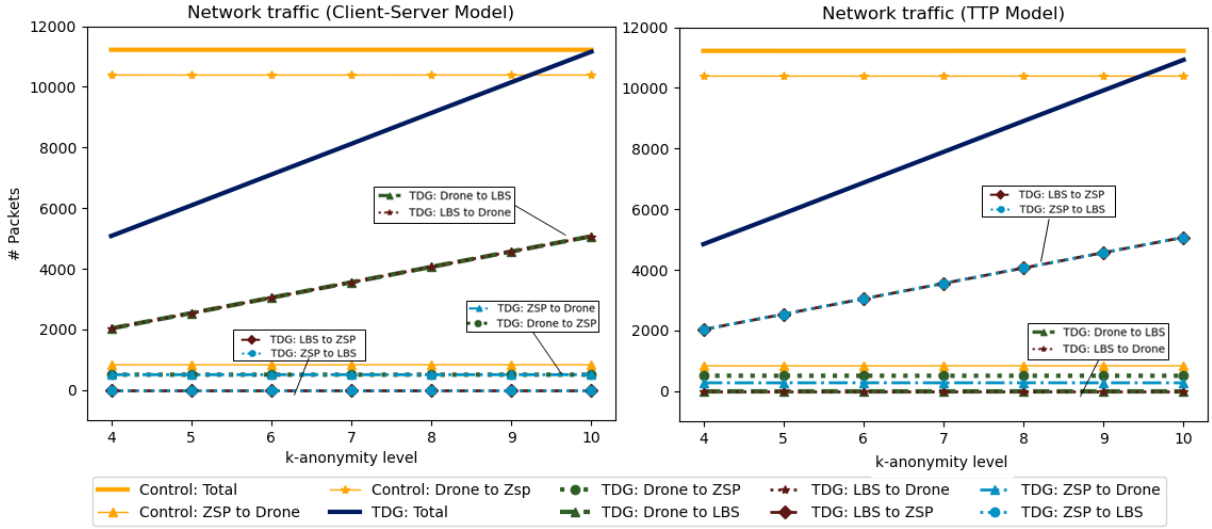
- **Number of packets:** it summarizes the number of generated packets from a given network source to a given destination. With this metric, we can analyze and identify what IoD nodes consume the most in the communication channel;
- **Real drone location discovery accuracy (RDLDA):** this metric measures the IoD-STIA success in terms of correctly associated queries. Let $\mathcal{Q}' = \langle q'_0, \dots, q'_n \rangle$ be the inferred group of queries by IoD-STIA, where n is the number of sequential queries of \mathcal{Q}' . Let $\mathcal{Q} = \langle q_{r_0}, \dots, q_{r_n} \rangle$ be the group of queries that corresponds to the drone's real queries. RDLDA can be calculated as follows:

$$RDLDA = \frac{\sum_{i=1}^n \begin{cases} 1, & q'_i = q_{r_i} \\ 0, & otherwise \end{cases}}{n}$$

- **Real trajectory time discovery rate (RTTDR):** it measures for how long the IoD-STIA can infer the drone's real trajectory, given the total drone's flight time t . Given two consecutive queries q'_i and q'_{i+1} , the real trajectory can be inferred if these queries correspond to q_{r_i} and $q_{r_{i+1}}$, respectively. RTTDR can be calculated as follows.

$$RTTDR = \frac{\sum_{i=1}^n \begin{cases} q'_{i+1}.t - q'_i.t, & q'_i = q_{r_i} \wedge q'_{i+1} = q_{r_{i+1}} \\ 0, & otherwise \end{cases}}{t}$$

Figure 4.13: Network traffic comparison regarding TDG performance evaluation



Source: Elaborated by the author

- **Distance rate per trajectory point ($DRTP$):** as important as to measure the drone's discovery is to know how distance difference IoD-STIA misses the correct location. We can evaluate this aspect through the similarity between trajectories. $DRTP$ highlights this aspect taking into account a distance rate per trajectory point, based on the TDG radius r . It is calculated as follows.

$$DRTP = \left(\frac{f(Q, Q')}{n} \right) / r$$

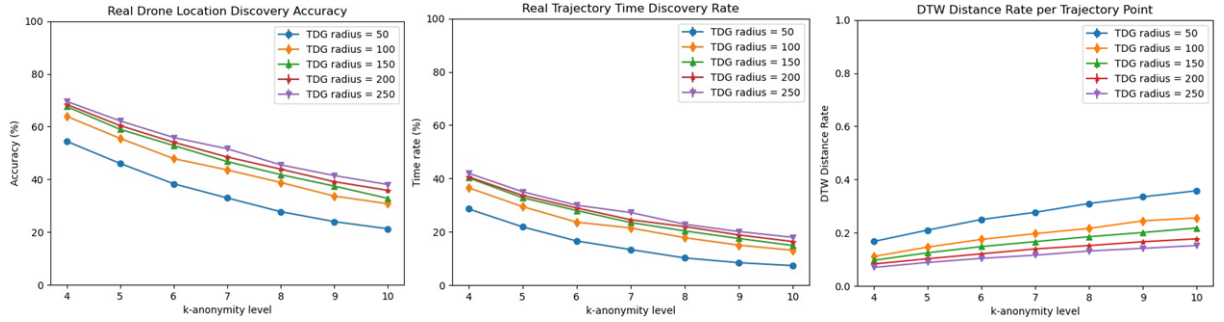
We consider as f the Dynamic Time Warping (DTW). It is a proper approach to calculate these trajectories' distances, since Q and Q' lead to trajectories with the same length and spatio-temporal coordinates at the same time [141].

4.4.5 Results and Discussion

Here we highlight the performance of TDG considering Client-server and TTP communication models, and the provided location privacy facing the IoD-STIA. Furthermore, for the majority of the experiments, the interval errors are less than 1%, being not represented in the charts.

- **Network Traffic Analysis:** The communication channel usage is directly related to the drone's SWaP constraints since the RF-based transmission consumes the drone's battery. Thus, the proposed mechanisms must use the wireless channel as little as possible.

Figure 4.14: Location privacy results regarding TDG performance evaluation



Source: Elaborated by the author

Figure 4.13 shows a network traffic comparison between both Client-server and TTP communication models for each k -anonymity level. The packets are analyzed in two main groups: control packets, related to flight and path planning messages; and TDG packets, related to the dummy-based mechanism.

As the simulations are replays with the same path planning for the same k value and seed, the number of control packets remains the same for all k -anonymity levels. On the other hand, the k level implies the number of generated dummy queries. Hence, the number of TDG packets increases whereas k also increases. In the Client-server model, this raise is slightly greater due to the request for topological information, generating more messages between the drones and the ZSP.

Due to the communication model's characteristics, there is no traffic between some components. In the client-server model, since drones communicate directly with the LBS, there are no exchanged packets between the LBS and the ZSP. In the TTP, in turn, the drones do not send the dummies to LBS directly since the ZSP is responsible for this task.

This leads us to the main difference between the models. We can observe a prominent communication between the drones and the LBS in the Client-Server model, which occurs through the available IoD wireless channel. In the TTP model, this prominence occurs between the LBS and the ZSP. This communication occurs through a wired channel since the ZSP commonly represents a grounded station. In any case, they will not use the same wireless channel shared by the drones.

► **Location Privacy Analysis:** the location privacy provided by a dummy-based LPPM commonly reflects three main aspects: the location information disclosure leverage by an attack, the time that the adversary does not infer the correct information, and the location information distance error. Here, they are evaluated through RDLDA, RTTDR, and DRTP, respectively. Figure 4.14 shows the obtained results for these three metrics. It is important to state that they are attack-centered metrics, in other words, it measures the IoD-STIA success.

As we can note, all metrics pointed out that the drone's real queries had more

privacy when we configured TDG with a lower radius value and a high k -anonymity level. *RDLDA* and *RTTDR* presented similar results, although they measured different aspects. In *RDLDA* with $R = 50$ m and $k = 10$, specifically, the attacker discovered less than 25% of the real drone's location during the time.

This same configuration provided the best result for *RTTDR* when the adversary inferred the drone's trajectory correctly less than 10% of the time. We also can note that for all configurations the attacker could not infer the real drone's trajectory more than 50% of the time. It highlighted the high location protection provided by TDG to potentially avoid trajectory-based attacks.

In the *DRTP* chart, higher rates indicated better location privacy since the trajectory distance was higher between the inferred and the real trajectory. Thus, this metric reinforced that constrained radius and higher k -anonymity levels provided better location privacy. However, *DRTP* pointed out that this distance was less than 40% of the configured radius, per trajectory point. Considering the best scenario, the *DRTP* was near 35% of the configured radius, more precisely, less than 20 m of the real drone's position.

► **General Discussion:** In summary, the experiments revealed that TDG could provide proper location privacy in the IoD environment, reducing the wireless communication channel utilization. Its application can be helpful in sparse scenarios since the proposed dummy generation method focuses on topology characteristics regardless of the presence of near drones.

TTP was the best-evaluated communication model to apply the TDG in the IoD paradigm. With this model, it is possible to reduce the wireless communication channel usage and mitigate the SWaP limitations, mainly the battery consumption. Regarding the location privacy analysis, TDG provided a high protection level facing IoD-STIA, whereas the configured generation radius was lower and the k -anonymity level was higher. More precisely, for a radius $r = 50$ m, the performance was truly better than the other ones, indicating that the proximity among the real and dummy queries increased the difficulty of an adversary to infer the optimal trajectory.

However, the evaluation also revealed a major shortcoming: considering the applied radius, the average distance error between the real and the inferred drone's location was small. It means that although IoD-STIA did not discover a significant amount of drone location and its trajectory, the inferred trajectory was not far from the real one. This issue poses new interesting challenges to further investigate, such as how the airways topology allied with the drones' trajectories affect these distances.

4.5 IoDAPM: An RL Approach for Dynamic Assignment of LPPMs

In the previous sections, we proposed t-MixDrones, MixRide, and TDG. Although all of them presented suitable levels of protection in terms of location privacy, they were designed for specific scenarios. Therefore, we can assume that no mechanism can be considered a “silver bullet” for addressing optimal location privacy. The high location privacy level commonly leads to a lack of QoS, such as high energy consumption and flight delay. Therefore, some research questions arise: (i) Which factors influence the LPPM effectiveness? (ii) How LPPMs affect the QoS of the network? (iii) How can the network assign the most suitable LPPM from a pool, considering the prevailing circumstances?

RL approaches represent suitable strategies to deal with these challenges, enhancing the knowledge about the dynamic behavior of a mobile network [142]. Also, these approaches can improve decision-making capabilities and adaptability in complex and dynamic environments. RL has been widely applied in different domains, ITSs and recommendation systems [143].

Considering these aspects, we propose **IoDAPM**, an RL-based approach for the **D**ynamic **A**ssignment of **P**rotection **M**echanisms in **IoD**. IoDAPM aims to improve the QoS provided by the network from the optimization of a transition model from the rewards obtained by previous assignments made in the network, considering the environmental conditions. The approach considers the ZSPs of each aerial zone as distributed agents that enhance a local RL-based model, which is updated globally periodically.

Specifically, we apply IoDAPM to assign dynamically our proposed LPPMs: t-MixDrones, MixRide, and TDG. However, our approach is not limited to these mechanisms in such a way that other further mechanisms can be considered. Furthermore, it is important to state that IoDAPM is not a LPPM, but an approach to select and apply the best mechanism given the environment conditions. Hence, the design of IoDAPM does not follow the proposed framework (Section 3.4).

In this section, we: discuss the fundamental aspects involved with RL approaches; formally define IoDAPM; present the performance evaluation of IoDAPM; and discuss the obtained results.

Table 4.9: Categorization of the IoD environment conditions

Condition	Categories
Drones' density	– drones per aerial zone $< D_{rate}$ (sparse) – drones per aerial zone $\geq D_{rate}$ (dense)
Battery level	– level $< 25\%$ (low) – level $< 50\%$ (medium) – level $\geq 50\%$ (sufficient)
Delay	– forecast $< 5\%$ (on time) – forecast $< 15\%$ (acceptable) – forecast $\geq 15\%$ (delayed)
Other networks (public buses)	– available – not available

4.5.1 Fundamental Aspects

RL consists of ML techniques in which agents interact with the environment to take actions that maximize a reward [143]. RL strategies can be applied to solve the assignment problems since it has some exploratory characteristics: the referred optimization delves into the interaction with the environment for better decision-making, considering a series of characteristics, starting from an unexplored scenario [144]. Formally, RL is modeled as a Markov decision process, composing a tuple $(\mathcal{S}, \mathcal{AT}, \mathcal{R}, \mathcal{P}, \gamma)$ such that \mathcal{S} represents all the possible states of the agent, \mathcal{AT} is the set of actions to take, \mathcal{R} is the reward function, \mathcal{P} represents the transition probability between states, and γ is a discount factor [144]. Moreover, the transition probability can follow a policy π such that π is optimized over time, aiming to maximize its cumulative rewards over time [142].

Besides location privacy, the proposed LPPMs cover different aspects. For instance, MixRid overcomes the energy issues of t-MixDrones. TDG, in turn, focuses on sparse environments considering the number of drones. It indicates that no ideal LPPM can provide optimal location privacy in any situation. Indeed, the mechanisms can provide enhanced levels of location privacy in different network conditions but also decrease other aspects, for instance, energy efficiency and causing delays. Table 4.9 categorizes the environmental conditions observed when these mechanisms were applied. It is important to state that these categories refer to a group of potential values for each condition. We discuss each one from a network point of view.

- **Drones density:** as in other mobile networks, the number of nodes (in this case, the drones) can severely affect the mechanisms, mainly the ones based on the k -anonymity principle. MixRide and t-MixDrones depend on a dense network to perform properly since their main concept delves into at least k nodes to be applied;
- **Battery level:** SWaP limitations still are a critical challenge in IoD. Therefore,

the usage of a given LPPM must consider the current battery level of the involved drones. For instance, t-MixDrones leads the drones to perform maneuvers to change their altitude, expending more energy than expected;

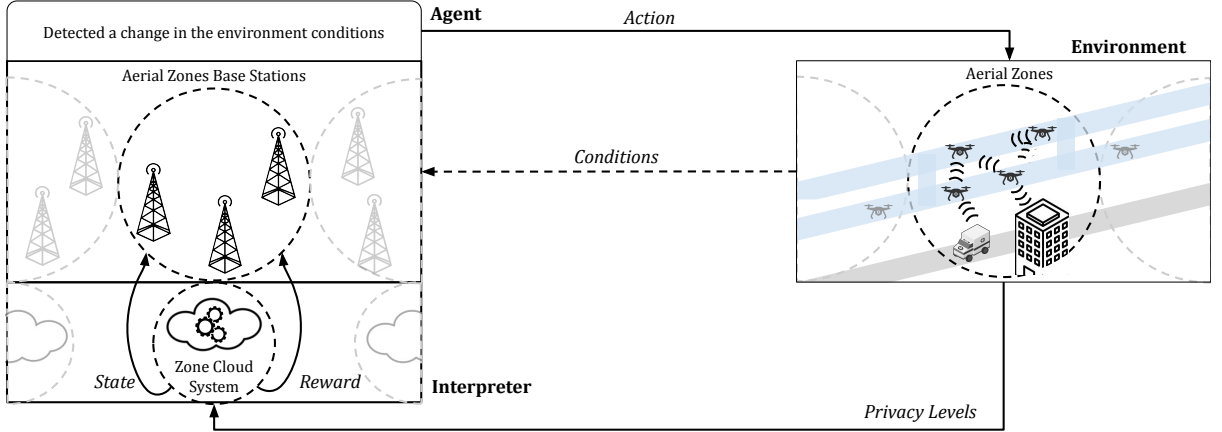
- **Delay:** likewise, drones must preserve a certain level of QoS regarding delay. Emergency services do not tolerate delays in the planned flight. Hence, LPPMs that can cause a delay in the drone’s flight – such as MixRide – may be applied only when the provided service is delay-tolerant.
- **Conditions from other networks:** although this condition is driven by the particular design of a given mechanism, the conditions of 3rd-party networks can affect the LPPM performance. As discussed, MixRide collaborates with grounded public transportation, and consequently, the provided privacy level depends on the public transportation conditions.

4.5.2 Design of IoDAPM

Based on the previous discussion, deciding which LPPM is the best approach to address an optimum location privacy level in a given environment is a noteworthy challenge. This problem can be defined as follows. Let us consider an IoD network with a navigation model G and a set of nodes \mathcal{N} . Also, let us consider a set $LPPM$ of available LPPMs to apply in the network. The dynamic assignment of IoD-based LPPM problem consists of assigning a mechanism $m \in LPPM$ over \mathcal{N} given a set of conditions \mathcal{C} extracted from the network considering a time interval Δ_t in such a way that m provides the highest privacy level $\forall m \in LPPM$, considering a set of metrics \mathcal{M} .

IoDAPM aims to solve the aforementioned problem. Its design has three parts. First, we define the main components of the approach. The two subsequent parts are related to the LPPM assignment step, and the global model update step, respectively. Figure 4.15 presents the concept of IoDAPM. It follows the traditional interface of RL models [143], consisting of the agent and environment entities, mediated by an interpreter gathering the environmental conditions. As the conditions can vary for different aerial zones, the proposed approach is applied over each aerial zone individually. Given the assignment of an available LPPM (action role) decided by the ZSPs of a given aerial zone (agent role), the IoD cloud system (interpreter role) observes the addressed privacy levels to define the reward and address the new state.

Figure 4.15: IoDAPM concept



Source: Elaborated by the author

► **Formal definition:** It is important to state that although the approach is valid for the whole IoD network as well as the environment, the definitions are centered on a given aerial zone $az \in \mathcal{AZ}$. Besides the common notation, we represent subsets with the following notation: given a set \mathcal{X} , \mathcal{X}_y represents a subset of \mathcal{X} . The **agent** is a subset $\mathcal{Z}_z \subset \mathcal{Z}$ such that every ZSP $z \in \mathcal{Z}_z$ is responsible for managing az . The **environment** corresponds to the entire region covered by az , embracing the network conditions of this region. The cloud system CS represents the **interpreter**, responsible for gathering the privacy levels and calculating the reward of the action taken, considering the reward function (Eq. 4.19).

The action actively consists of assigning an LPPM given the environmental conditions. Hence, the set of actions \mathcal{Act} is composed of a single action α , representing the assignment. The *Conditions* refer to a set of environmental characteristics that can affect the performance of a given LPPM. Formally, \mathcal{C} represents the set of conditions such that $\forall c \in \mathcal{C}$, c can assume a discrete value from a group of condition categories, as presented in Table 4.9.

The set of possible *states* \mathcal{S} corresponds to all tuples coming from the combination of $LPPM \times \mathcal{C}_c$, such that \mathcal{C}_c is the set of all possible tuples considering the combinations of the observed environmental conditions $\mathcal{C}_0 \times \mathcal{C}_1 \times \dots \times \mathcal{C}_n$, such that n is the number of categories of the environmental conditions. For instance, considering Table 4.9, there are four different categories. A possible tuple from \mathcal{C}_c can be (“sparse”, “low”, “on time”, “available”), which in turn can be combined with “TDG” to compose a valid tuple in \mathcal{S} .

Furthermore, IoDAPM considers the provided QoS as a function, measured by a set of metrics \mathcal{M} and a set of weights \mathcal{W} , $|\mathcal{M}| = |\mathcal{W}|$. Hence, the QoS provided by the network at a time interval Δ_t is defined by the following equation.

$$\text{QoS}(\Delta_t) = \sum_{i=0}^{|\mathcal{M}|} w_i m_i(\Delta_t), w_i \in \mathcal{W}, m_i \in \mathcal{M} \quad (4.17)$$

► **Policy:** representing the states \mathcal{S} through the previous combinations can lead to a higher number of possible states, representing a high-dimensional problem [144]. For instance, considering the three related IoD-based mechanisms and all the possible categories of each environmental condition (Table 4.9), the RL approach has 108 possible states. However, given the observed environmental conditions, the possible transitioning states are constrained by combining the LPPMs with these conditions. For instance, let us suppose the tuple $obsConds=(\text{“sparse”}, \text{“low”}, \text{“on time”}, \text{“available”})$ as representing the observed conditions. The next state $s_{next} \in \mathcal{S}$ will be obligatorily ($\text{“TDG”}, obsConds$), ($\text{“MixRide”}, obsConds$), or ($\text{“t-MixDrones”}, obsConds$). Therefore, the agent addresses a next state s_{next} respecting the policy π defined as follows:

$$\pi = s_{next}, \exists s_{next} \in \mathcal{S}_{obs}, \mathcal{S}_{obs} \subset \mathcal{S} \text{ and } \mathcal{S}_{obs} = LPPM \times obsConds \quad (4.18)$$

In other words, the next state is always assigned randomly from a set of $|LPPM|$ possibilities, reflecting the last observed conditions of the environment. Although this assignment occurs randomly, it follows a transition probability and a reward model defined by a learning model, discussed as follows.

► **Reward:** a key aspect of RL-based approaches is measuring how good the action taken by the agent was. This measurement varies according to the approach, considering the particularities of the problem to be solved. In IoDAPM, the reward is stored in a matrix \mathcal{R} with dimensions $|\mathcal{S}| \times |\mathcal{S}|$. Let us assume (i, j) as the index of \mathcal{R} related to the last action taken, Δ_t as the time interval related to this action in the aerial zone, and Δ_{t-1} the time interval of the predecessor action taken. A function calcReward calculates the reward related to this action, being formally defined in Eq. 4.19.

$$\text{calcReward}(i, j, \Delta_t, \Delta_{t-1}) = \mathcal{R}(i, j) + (\text{QoS}(\Delta_t) - \text{QoS}(\Delta_{t-1})) \quad (4.19)$$

The reward is based on the difference in QoS between the two last actions taken. If the previous provides a QoS level worse than the one provided by the predecessor, it is a punishment, decreasing the last attributed reward value. Otherwise, the action has enhanced the network, improving the associated reward.

► **Transition Probability:** the transition probability refers to a model represented by a matrix \mathcal{P} with the same dimension as \mathcal{R} such that $\mathcal{P}(i, j)$ is the probability from the agent transitioning from the state s_i to s_j , such that $s_i, s_j \in \mathcal{S}$ and i, j are indexes related to these states. IoDAPM enhances this model based on the knowledge acquired from the environment. Therefore, the attribution of transition probabilities has two distinct phases: the initialization step, corresponding to the baseline model; and the enhancement step, updating \mathcal{P} based on the conditions and the provided QoS.

Algorithm 9: Calc-Transition-Prob

Input : $\mathcal{P}, \mathcal{R}, i_{\text{last}}, j_{\text{last}}, \mathcal{I}_{\text{last}}$

- 1 $lastProb \leftarrow \mathcal{P}(i_{\text{last}}, j_{\text{last}})$
- 2 $\mathcal{P}(i_{\text{last}}, j_{\text{last}}) \leftarrow \max(0, \min(1, lastProb + (\mathcal{R}(i_{\text{last}}, j_{\text{last}}) * lastProb)))$
- 3 $remainedProb \leftarrow (\mathcal{P}(i_{\text{last}}, j_{\text{last}}) - lastProb) / |LPPM|$
- 4 **for** $(i, j) \in \mathcal{I}_{\text{last}}$ **do**
- 5 **if** $(i, j) \neq (i_{\text{last}}, j_{\text{last}})$ **then**
- 6 $\mathcal{P}(i, j) \leftarrow \max(0, \mathcal{P}(i, j) - remainedProb)$

Output: \mathcal{P}

In the initialization step, the main idea is to initialize equally the probability of transitions from a given sort of condition tuples and the available mechanisms to assign. Let us assume \mathcal{P}^0 as the initial transition probability model. Considering the policy π , the transition from a state s_i to a state s_j has $|LPPM|$ possibilities. Hence, it is intuitive that $\forall (i, j)$ of \mathcal{P}^0 , $\mathcal{P}^0(i, j) = \frac{1}{|LPPM|}$.

The enhancement step aims to optimize \mathcal{P} so that the transition probabilities adapt to the QoS observed over time based on previously taken actions. Therefore, this step occurs periodically, being processed before the agent takes a new action. Let us assume $(i_{\text{last}}, j_{\text{last}})$ as the matrix index of the last transition taken by IoDJAPM, and $\mathcal{I}_{\text{last}}$ a set containing the states' indexes that could be reached in the last assignment. The enhancement step is given by Algorithm 9.

An important aspect of Algorithm 9 is the need to update the transition probabilities of a selected group of elements instead of the whole matrix, referring to the constrained set of possible transitioning states. There are two cases covered in the enhancement: (i) the update of the transition whose index refers to the last transition taken (Lines 1–3); and (ii) the probability updates of the remaining states that could be assigned but were not taken (Lines 4–6). The updated probability of the last transition is calculated by summing the multiplication of the associated reward by its current probability. If the reward is a punishment, the probability is decreased. To keep the transition probabilities concise, the value is constrained in the interval $[0,1]$ (Lines 1–2). Since this new probability affects the remained states in the group, they are adjusted with the remained probability (Line 3), being equally distributed over the other states that could be reached (Lines 4–6).

► *Time Complexity Analysis:* Algorithm 9 is dominated by matrix update iterations, in terms of time complexity. From Line 1 to 3, we have basic operations whose data are gathered from the matrices \mathcal{R} and \mathcal{P} , corresponding to a complexity of $\mathcal{O}(1)$. From Line 4 to 6 occurs the transition probability matrix update, leading to a complexity of $\mathcal{O}(|\mathcal{S}|^2)$. However, in practice, the updating is bounded by the states that could be reached in the assignment, corresponding to $\Theta(|LPPM|)$ since the possible transitions are guided by the number of mechanisms to assign.

Algorithm 10: Assign-LPPM

Input : $LPPM, obsConds, \mathcal{S}, s_{curr}, \mathcal{P}, \Delta_{t-1}$
Output: $\mathcal{S}, \mathcal{R}, s_{new}$

- 1 $\Delta_t \leftarrow \text{currentTime}()$ - the end time of Δ_{t-1}
- 2 $i, j \leftarrow$ get the related index of s_{curr} considering \mathcal{S} and \mathcal{R}
- 3 $lastConds \leftarrow$ extract the last conditions, associated with s_{curr}
- 4 $\mathcal{R}(i, j) \leftarrow \text{calcReward}(i, j, \Delta_t, \Delta_{t-1})$ ▷ Eq. 4.19
- 5 $\mathcal{S}_{last} \leftarrow LPPM \times lastConds$
- 6 $\mathcal{I}_{last} \leftarrow$ generate a set containing the indexes in \mathcal{S} that correspond to the states of \mathcal{S}_{last}
- 7 $\text{Calc-Transition-Prob}(\mathcal{P}, \mathcal{R}, i, j, \mathcal{I}_{last})$ ▷ Alg. 9
- 8 $\mathcal{S}_{updt} \leftarrow LPPM \times obsConds$
- 9 $s_{new} \leftarrow$ apply π over \mathcal{S}_{updt} ▷ Eq. 4.18

► **Dynamic assignment of LPPM strategy:** Algorithm 10 describes the IoDAPM assignment performed in the CS of each region. It is activated each time a new set of environmental conditions is observed in the aerial zone. It requires, as input, the set of LPPMs, the tuple $obsConds$ representing the observed conditions of the environment, the current state s_{curr} , and the last observed time interval Δ_{t-1} . As output, the algorithm provides the updated matrices of transition probability \mathcal{S} and reward \mathcal{R} , and the attribution of a new state.

Initially, the algorithm calculates the time interval considering the last assignment (Line 1) and gets the index of the current state considering the model of the matrices \mathcal{R} and \mathcal{S} (Line 2). Also, the environmental conditions of the last assignment are taken (Line 3). With these data, the reward of the current state is updated, following Eq. 4.19 (Line 4). After, the algorithm proceeds with the update of the transition probability matrix \mathcal{S} following Algorithm 9 (Lines 5-7). It is important to note that these updates always “look back” to the impact of the last mechanism assignment. With both matrices \mathcal{R} and \mathcal{S} updated, the algorithm finally calculates the new assignment, following the policy π , as defined in Eq. 4.18 (Line 9). The new current state is communicated to all ZSPs of the aerial zone, which in turn must update the mechanism applied in the region.

► *Time Complexity Analysis:* Algorithm 10 presents several constant operations (Lines 1, 3-5, 8, 9) whose complexity is $\mathcal{O}(1)$. Likewise Algorithm 9, matrix iterations in \mathcal{S} , \mathcal{R} , and \mathcal{P} dominate the time complexity (Lines 2, 6, 7), also leading to $\mathcal{O}(|\mathcal{S}|^2)$. However, in practice, the related base stations compute just a small part of the matrices, being also $\Theta(|LPPM|)$.

► **Global model update strategy:** Algorithm 10 generates different models for the matrices \mathcal{R} and \mathcal{S} since each aerial zone can be considered an individual agent. Therefore, aerial zones can benefit from an enhanced model processed and shared globally, increasing location privacy in all regions as they share the same communication channel.

We design an update strategy that merges the models from different aerial regions,

Algorithm 11: Update-Models

Input : $\mathcal{S}_z, \mathcal{R}_z, LPPM, obsConds$
Output: $\mathcal{S}_{best}, \mathcal{R}_{best}$

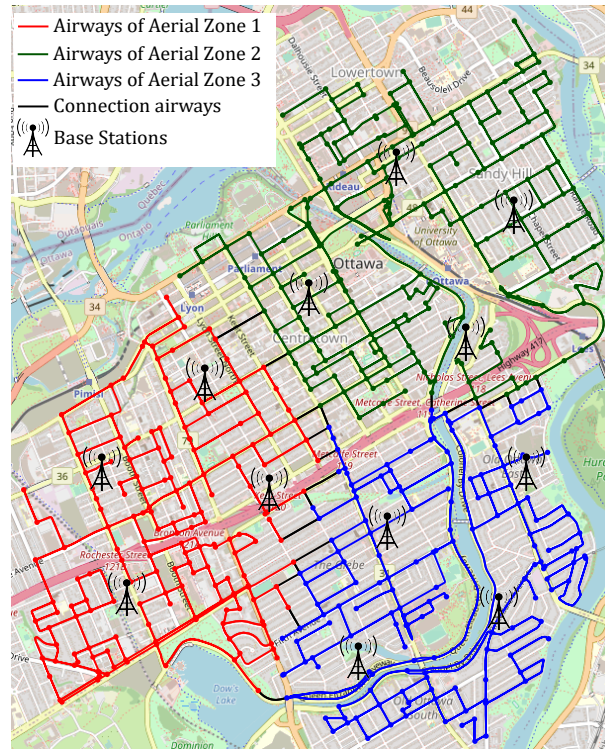
- 1 $\mathcal{S}_{best}, \mathcal{R}_{best} \leftarrow$ empty matrices with dimensions of $\mathcal{S}_z, \mathcal{R}_z$
- 2 $\mathcal{I}_{conds} \leftarrow$ group the indexes of \mathcal{R}_z considering the distinct combinations of $LPPM \times obsConds$
- 3 **for each** group of \mathcal{I}_{conds} **do**
- 4 $bestReward \leftarrow 0$
- 5 $bestRegion \leftarrow \emptyset$
- 6 **for each** \mathcal{R} of \mathcal{R}_z **do**
- 7 $sumReward \leftarrow \sum_{(i,j) \in group} |\mathcal{R}(i,j)|$
- 8 **if** $sumReward \geq bestReward$ **then**
- 9 $bestReward \leftarrow sumReward$
- 10 $bestRegion \leftarrow$ region az associated with \mathcal{R}
- 11 update $\mathcal{S}_{best}, \mathcal{R}_{best}$ in the indexes of $group$ with the values from \mathcal{S}, \mathcal{R} associated with $bestRegion$

extracting the best rewards of each one to compose an enhanced model. Algorithm 11 describes this strategy. It requires as input the probability transition and reward matrices of all aerial zones (\mathcal{S}_z and \mathcal{R}_z , respectively), the set of LPPMs, and the tuple of observed conditions $obsConds$. As output, the algorithm provides the best models \mathcal{S}_{best} and \mathcal{R}_{best} .

The main idea of the strategy is to discover what aerial zone has the best reward so far for each group of transitions. Hence, the algorithm initializes the best models as empty matrices (Line 1) and groups the matrix indices considering these groups of transitions (Line 2). After, the evaluation of each group of transitions begins (Lines 3-11). The reward matrices \mathcal{R}_z of the aerial zones are analyzed one by one (Lines 6-10) such that the matrix with the greatest reward (absolute value) sum is taken as the best region for the group. This strategy is based on the premise that the highest rewards (or punishments) contribute more effectively to the learning model. After evaluating all the matrices, the algorithm updates the values of the generated best matrices in the indices related to the given group (Line 11). After, both \mathcal{S}_{best} and \mathcal{R}_{best} must be sent to the cloud system of all the aerial zones in such a way they replace the local models.

► *Time Complexity Analysis:* differing from the previous algorithms, Algorithm 11 computes the entire related matrices (Line 1), leading to a time complexity of $\mathcal{O}(|\mathcal{S}|^2)$. The iteration block (Lines 3-11) goes through each group of conditions, leading to a complexity of $\Theta(|LPPM| \times |\mathcal{C}|)$. Considering that $|\mathcal{S}| = |LPPM| \times |\mathcal{C}|$, the initialization of the matrices is dominant in terms of time complexity. Therefore, the time complexity of Algorithm 11 is $\mathcal{O}(|\mathcal{S}|^2)$.

Figure 4.16: Simulated environment regarding IoDAPM performance evaluation



Source: Elaborated by the author

4.5.3 Simulation Setup and Performance Evaluation

To analyze the performance of IoDAPM, we carry out a performance evaluation comparing it with the three related LPPMs applied individually: TDG, t-MixDrones, and MixRide. With this evaluation, our main goal is to compare the performance of the approaches considering the provided QoS and the aspects that affect it, namely: the location privacy level, the drone's energy consumption, and the potential flight delay. However, IoDAPM is not limited nor dependent on these aspects, and other arrangements can be considered.

As discussed, the problem tackled in this study has an exploratory behavior. Hence, simulations of the IoD environment are a suitable way to carry out this evaluation, exploring the different network conditions as much as possible. To perform the simulations, we use IoDSim. The performance of IoDAPM is directly related to the exploration of the environment and, therefore, to the observation of different network conditions. Hence, it is intuitive that both reward and transition probabilities models demand constant improvement until they represent proper models to provide a suitable QoS. Bearing these aspects in mind, we divide our evaluation into two fronts: (i) we carry out a training phase of IoDAPM, aiming to obtain enhanced models; and (ii) the comparative evaluation between the mechanisms.

Table 4.10: IoDAPM Setup

Parameter	Value
LPPMs	{tMix, MixR, TDG}
Env. Conditions	According to Table 4.9
Set of metrics \mathcal{M}	{ C_{rate} , TMA , TDR , PCR }
Set of weights \mathcal{W}	{0.1, 0.4, 0.25, 0.25}
Initial state	{TDG, sparse, sufficient, on time, not available}

► **Simulation Setup:** we consider a simulated environment where drones perform a general urban mission service, communicating with each other and ZSPs. Drones have an associated path planning, defined before they start flying. We model the region of Ottawa downtown (Canada) as the urban environment, covering an area of about 18 km², illustrated in Figure 4.16. In this scenario, we consider three different aerial zones (highlighted by different colors). Each aerial zone has four ZSPs to manage both the communication and navigation services. There are two available altitudes to fly: 50 and 100 m. Due to privacy concerns, the shared airways are logically deployed parallel to the ground roadways [2, 5]. To support MixRide, we simulate the city’s bus routes provided by the OC Transpo company².

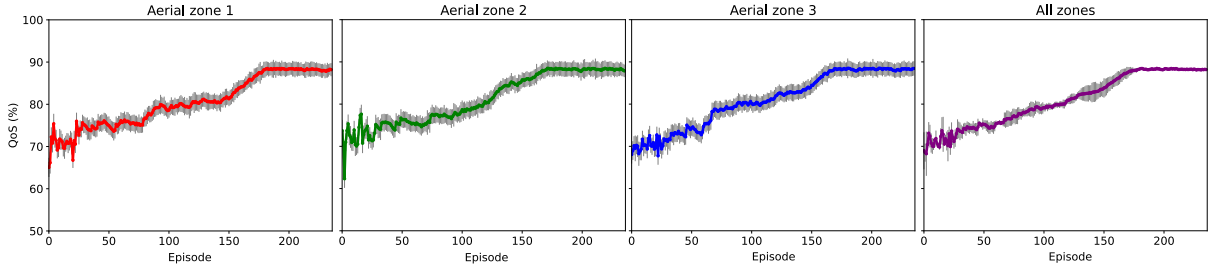
We consider 100 drones flying over the environment. They are randomly spread over the aerial zones, varying their initial position in each simulation. Also, they fly following defined path plannings, guided by the ZSPs. The mobility pattern is based on Gauss-Markov with an average speed varying from 5 to 10 m/s. Their energy consumption model follows the drone MD4-300, whose expected time flight is about 45 minutes [20]. There are 18 buses available such that they follow three different routes related to the Ottawa downtown region (routes 56, 14, and 6). Regarding the communication model, IoD nodes have the same protocols and radio configurations, using TCP, AODV, and CSMA/CA. The radio operates following the standard 802.11n with modulation mode of 1×1 20 MHz, having a maximum throughput of 72.2 Mbps.

Regarding the LPPMs, t-MixDrones deploys four Mix Zones per aerial zone and has a k -anonymity set as 4, while in MixRide is 3 (defining the maximum number of rides per bus). Also, MixRide is set with configuration C2, balancing both the delay and power consumption. All mechanisms are set with a coverage radius of 250 m. Besides the presented setup, each LPPM follows the configuration with its best performance, according to the performance evaluation presented in this chapter for each mechanism, accordingly.

Table 4.10 shows the setup of IoDAPM for the training phase and the comparative evaluation. Besides the three related LPPMs and the conditions defined in Table 4.9, we set the following arrangement as the initial state of IoDAPM for all aerial zones: we take

²Available at: <https://www.octranspo.com/en/plan-your-trip/travel-tools/developers/>

Figure 4.17: Average Quality of Service (QoS) per aerial zone in the training phase



Source: Elaborated by the author

TDG as the initial mechanism since it does not depend on a minimum number of drones in the nearby region to apply, sparse drones' density, sufficient battery, trips on time, and not available public buses.

► **Metrics:** the metrics used to calculate the QoS are divided into location privacy-related and general drone service aspects. These metrics are already defined in the previous sections, being: C_{rate} , TMA , TDR , and PCR . The applied weights aim to prioritize equally the two fronts of metrics. Regarding the weights of location privacy-related metrics, TMA has a greater weight than (C_{rate}) because it represents the success of an attacker, indicating a lack of location privacy, indeed.

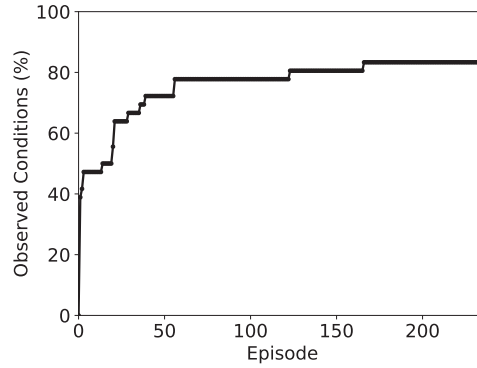
4.5.4 Results and Discussion

This section presents the results of both the IoDAPM training phase and the comparative evaluation. We discuss the evaluation insights, highlighting the advantages and challenges faced by IoDAPM.

► **IoDAPM training phase:** starting from the setup presented in Table 4.10 and with the initialization models \mathcal{R}^0 and \mathcal{P}^0 , we carried out the training phase of IoDAPM. Each iteration had a simulation time of 3 hours, and the best models of an episode e were the input models for the iteration $e + 1$. The stopping criteria consider two aspects: a maximum number of 1000 episodes, or a QoS convergence in the last 100 episodes. In the latter, we assume the convergence when the standard deviation $\sigma(QoS) > 1$.

Figure 4.17 presents the QoS average per aerial zone and the entire network. The gray vertical bars indicate the interval error since each value corresponds to the average of each simulation (3 hours). The training stopped due to the QoS convergence after 237 episodes. Starting from a QoS of about 65% in each zone, the models were improved,

Figure 4.18: Cumulative rate of observed conditions per episode



Source: Elaborated by the author

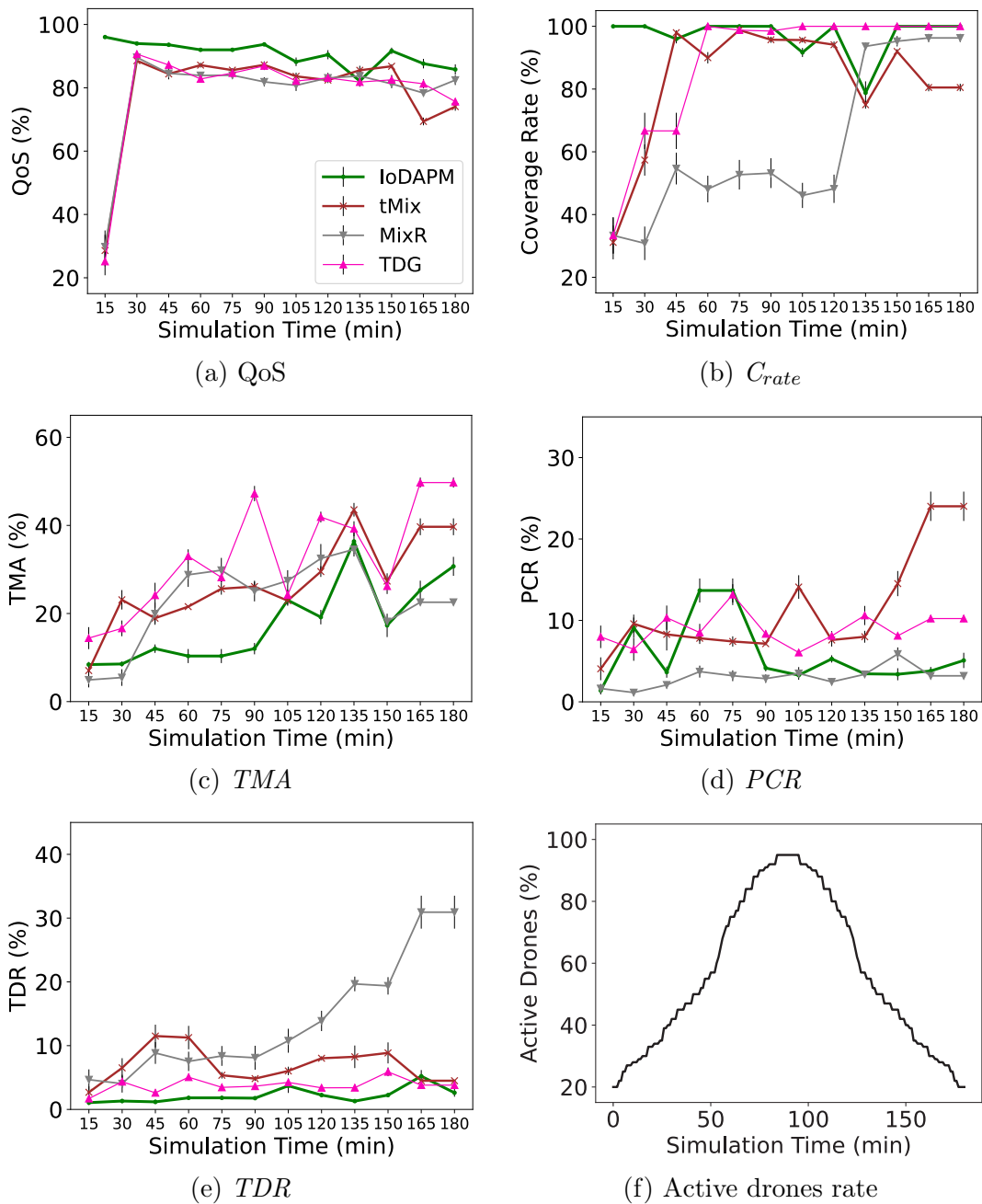
addressing a QoS average of 88.35 % in the last episode. Considering that drones have an inherent power consumption and the potential success of an attacker to track the drone’s trace (even at a minimum level), the addressed QoS pointed out that both the resulting \mathcal{R} and \mathcal{P} models are robust enough to assign LPPMs dynamically in IoD.

One of the main advantages of the training phase is the exploration of new observed conditions, presented in Figure 4.18. We note rapid, observable conditions increasing in the first quarter of episodes. Comparing this aspect with the curve of QoS of each aerial zone (Figure 4.17, it is significant that a continuous “adaptation” of the models such that they were handling the new observed conditions. When the discovery of new observed conditions stabilizes, IoDAPM can enhance the models thoroughly. The ascending curve of QoS can note this aspect until their convergence in the last quarter of episodes.

► **Comparative evaluation:** considering the enhanced models from the training step, we perform a comparative evaluation between IoDAPM and the related LPPMs. We carried out 35 distinct simulations with different path planning for drones, where they perform missions for 3 hours. These simulations were replicated for each compared mechanism, totaling 140 simulations.

Figure 4.19 presents the results of the simulations considering five different metrics: the QoS, calculated according to Eq. 4.17 (Figure 4.19a); C_{rate} (Figure 4.19b); TMA (Figure 4.19c); PCR (Figure 4.19d); and TDR (Figure 4.19e). We extend our concept of QoS to the related LPPMs in such a way that this metric is also calculated following the former metrics applied with the corresponding weights (Table 4.10). In these charts, each value corresponds to the observable metric level in the past 15 minutes, considering the average of the 35 simulations. For instance, the QoS value at 60 min in Figure 4.19a corresponds to the obtained QoS average from 45 to 60 min. We also present the rate of active drones over time, considering all the aerial zones (Figure 4.19f). This rate grows until half of the simulation time and then starts to decrease (drones closer to the destination are forced to land). With this behavior, a wide range of conditions are

Figure 4.19: Results of each considered metric of the comparative evaluation



Source: Elaborated by the author

observed in the same simulation, enhancing the proposed evaluation.

The results highlight a key advantage of IoDAPM: the approach provides the highest QoS level regardless of the environmental conditions. Except for an anomaly noted during the time interval from 120 to 135 minutes, IoDAPM outperforms all the compared mechanisms, addressing a QoS higher than 85%. Although the compared mechanisms present a QoS near 82% in most situations, none of them has a prevalence, varying its performance according to the network conditions. We can note that this variation is stronger at the beginning and the end of the simulation, where the number of active

drones is growing and decreasing faster, which promotes a very dynamic environment.

Considering the other metrics individually, IoDAPM presents a “stable” behavior. While the compared mechanisms vary significantly over time, IoDAPM prevails with a smooth curve. For instance, MixRide has a lower coverage rate in the first half of simulations because drones are sparse in the beginning, and the conditions to apply the ride are not satisfied. The coverage increases significantly in the second half when the drones can take the ride. However, the associated delay also increases since the bus generally moves slower than drones. Similar behavior occurs with TDG regarding the TMA . Although TDG provides a broader C_{rate} , its protection is strictly related to the airways topology, varying significantly depending on the region the drone requests the dummy queries. Regarding t-MixDrones, the PCR increases at the end of the simulation when the number of drones decreases. Observing this result in detail, the high PCR occurred due to the remaining drones having a low battery charge. As t-MixDrones requires additional maneuvers, the battery level (which was already low) ended up draining more, following the established energy model [20].

Although IoDAPM presented suitable levels of QoS, some challenges emerged. We can note a significant decrease in the time interval from 120 to 135 minutes, which was caused by a lower C_{rate} and higher TMA . Observing the IoDAPM behavior accurately at this time, the approach had assigned TDG as the mechanism in the majority of simulations, which was a reasonable assignment since the number of drones was decreasing. However, the drones did not request dummy queries at the time interval, leading to the presented results. Another anomaly refers to the PCR level from 45 to 75 minutes, being caused by successive airway changes of t-MixDrones assignments. Nonetheless, it did not affect the QoS in general. Therefore, new enhancements must be planned in future work to mitigate these situations.

4.6 Case Study: Impact of Remote ID Rule in the Drone’s Location Privacy

While drones began to be used in several areas, there was also a significant increase in their unauthorized use. The FAA reported 2,595 unauthorized drone operations over the United States territory in 2021, representing an increase of 59% compared to 2020 [145]. Concerning these issues, the FAA published the Remote ID, a rule dedicated to UAVs [91]. Any drone with a Remote ID must broadcast its sensitive information while operating over the airspace [91]. Hence, authority devices can monitor the airspace, and,

in case of accidents or unauthorized access, the authorities can take reasonable measures. This rule was discussed in detail in Section 3.2.7. Furthermore, other drones can be aware of their neighborhood. The deployment of Remote ID is already underway in the United States [145], where all drone pilots required to register their UAS must operate their aircraft following the rule beginning in September of 2023. Also, drone manufacturers must comply with the requirements in their devices until September 2022.

Although the primary goal of Remote ID is to provide a safer environment for the drones, this rule can be a threat to drone's privacy. In the final rule of Remote ID, the drone's personal information must be broadcast as clear text. Any device with the Remote ID module can receive the data. From an attacker's point of view, Remote ID allows a malicious entity to obtain enhanced knowledge about the drones since the attacker has a device with the Remote ID broadcast module. Location-based attacks, for instance, can gather a drone's location information and apply some techniques to obtain knowledge of the drones' mobility patterns and, therefore, about their location and identity. Besides revealing these data, some attacks try to track the full (or a part of) node's trajectory [9]. When we look at the IoD envisioned scenario, the Remote ID deployment represents the sharing of sensitive data not only from drones, but also from the provided services, from companies, and even from the users of these services.

With this case study, our main goal is to technically demonstrate that Remote ID is a threat to the drone's location privacy in the IoD environment. Also, we aim to analyze if existent protection mechanisms for IoD can be designed as an extension of the Remote ID to provide an adequate level of location privacy. To address this goal, we simulate an IoD environment with eavesdroppers spread over the ground intercepting the Remote ID messages. We perform a location-based attack through these collected data, tracking the drone's trajectory.

Our contributions are threefold:

- ▶ We design a location-based attack considering the Remote ID message specification, focusing on tracking the drone's trajectory;
- ▶ Through extensive simulations, we technically demonstrate that Remote ID is a serious threat to the drone's location privacy in the IoD, where more than 90% of the drone's trajectory can be tracked;
- ▶ We also demonstrate that existing mechanisms can enhance the standard Remote ID protocol, mitigating the attack success and providing a better level of location privacy for drones. Specifically, this study analyzes the ARID [146] and t-MixDrones (defined in Section 4.2.6) protection strategies.

4.6.1 Related Proposals

As Remote ID is a recent specification, few studies investigated this new rule. Furthermore, most of them examined different technologies to deploy the Remote ID. For instance, LoRaWAN [147], Bluetooth and WiFi [148], and Globally Unique Flight Identifier (GUFID) [149]. Regarding security and privacy, Tedeschi et al. [146] proposed a pseudonym-generation strategy, ARID, which aims to enable a drone to generate Remote ID messages that can be verified only by legitimate authorities. The drone’s pseudonym changes for any message due to dependence on the timestamp data. Although the authors discussed that their solution could provide partial protection against location-based attacks, they did not present formal proof in this regard.

The LPPMs proposed in this dissertation, although did not focus on the Remote ID rule, have the potential to enhance the FAA proposal. For instance, t-MixDrones can be applied together with Remote ID or ARID [146], representing a potential enhancement. In this case study, we integrate t-MixDrones and ARID [146]. Our goal is to mitigate the attacker’s success, providing a better level of location privacy for IoD compared to the standard Remote ID protocol.

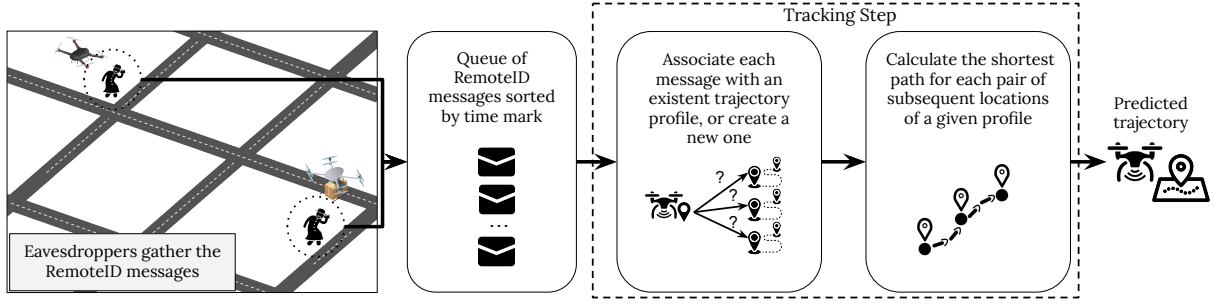
4.6.2 Design of Remote ID Location-based Attack

In this section, we design a location-based attack focusing on the Remote ID protocol. Firstly, we define some assumptions related to the considered application scenario:

- ▶ The mobile network infrastructure follows an IoD paradigm where the flyable airspace is represented as a graph $G = (V, E)$, as discussed in Section 2.2;
- ▶ Zone Service Providers (ZSPs) control the airspace, giving fair and shared access to drones;
- ▶ Drones fly over the defined airways $G.E$ strictly. Furthermore, every drone has a path planning when flying, with a well-defined destination point. Also, both the takeoff and landing must occur on a node $v \in G.V$;
- ▶ All the IoD nodes have a built-in standard remote ID module, enabling them to send and receive these messages. We assume a dedicated RF spectrum to broadcast them, differing from the one used for managing the IoD network;

Given the assumptions mentioned above, we can formalize some supplemental elements involved in this environment:

Figure 4.20: Concept of the designed Remote ID Location-based Attack



Source: Elaborated by the author

► \mathcal{M}_{RID} is the set of Remote ID messages broadcast in the environment. $\forall m \in \mathcal{M}_{RID}$, m contains all the related data of a standard message;

► \mathcal{M}_A is the set of Remote ID messages eavesdropped by \mathcal{A} such that $\mathcal{M}_A \subseteq \mathcal{M}_{RID}$;

► Each involved drone d broadcasts a message m following a time interval Δ_t , such that $\Delta_t \leq 1$ second. Also, the built-in Remote ID module has a predefined transmission range r , described in meters;

► The adversaries \mathcal{A} have a background knowledge \mathcal{K} about the IoD topology (e.g., how the airways are spread and how many altitudes are available to fly) since the airways follow the terrestrial roads, mostly;

The main goal of each adversary $\alpha \in \mathcal{A}$ is to eavesdrop on the Remote ID messages from the drones, sharing them with all the attackers. Hence, \mathcal{M}_A represents a spatiotemporal dataset of the drones' information. This dataset is the input to the location-based attack, defined in the next section.

To track a given drone's trajectory, we design a location-based attack definition based on the assumption that, given a source and a destination, the drone will fly following the shortest path between these points (introduced firstly in Section 4.2.1).

► Let us consider a drone $d \in \mathcal{D}$, a set \mathcal{M}_{Ad} representing the eavesdropped messages related to d , and \mathcal{C}_M a sequence of spatio-temporal coordinates from \mathcal{M}_{Ad} , sorted by the timestamp of each message $m \in \mathcal{M}_{Ad}$. Also, let us consider a function $F(c_s, c_d, G, \Gamma)$ such that F calculates the shortest path $\mathcal{P}_{shortest}$ between two coordinates c_s and c_d , based on a set of metrics Γ over the G topology. The aggregation of each shortest path $\mathcal{P}_{shortest}$ generated by the application of F over all coordinate pairs $\langle c_i, c_{i+1} \rangle$ from \mathcal{C}_M represents the d 's tracking trajectory \mathcal{P}_{track} .

In other words, for each subsequent pair of messages $\langle m_1, m_2 \rangle$ eavesdropped from a drone, the attack assumes that the drone flew following the shortest path from the broadcast of m_1 until the broadcast of m_2 . This is a very reasonable assumption if we consider that drones have SWaP (Size, Weight, and Power) limitations.

Considering the Remote ID final's rule, the drone's location is a trivial task to an

attacker because the drone broadcasts this data as plaintext [91]. On the other hand, with anonymization-based mechanisms, the drone’s location is harder to track as their identities change over time. In both cases, the complete tracking of the drone’s trajectory depends on three factors: the broadcast frequency; the message’s transmission range; and the strategic positioning of the eavesdroppers.

Apart from all these aspects, we design a location-based attack whose input data are the Remote ID messages. This attack occurs in real time, instantly processing the messages and tracking the locations. Figure 4.20 depicts its concept. Different eavesdroppers, distributed over strategic locations in the environment, gather the Remote ID messages broadcast by drones. These messages are sent to a unified system that stores them in a queue \mathcal{M}_A , sorted by the time mark. The tracking stage begins from the UA ID and location message field, where the attacker tries to associate the message with an existent trajectory profile.

Algorithm 12 describes the tracking step of our designed attack, being called to process every message $m \in \mathcal{M}_A$. Also, the algorithm requires as input the topological airways graph G and the set of trajectory profiles T . Each profile has a sequence of waypoints P that represent the profile trajectory, a velocity value v related to the UA velocity data of the last Remote ID message associated with the profile, and a time mark t . We consider an error ϵ related to the velocity variance to define the best profile for m .

Initially, the algorithm verifies if there is some trajectory profile τ already associated with the ID described in m (Line 4). The affirmative case indicates no anonymization mechanism is applied in the current Remote ID messages, which means that the protocol follows the final rule’s proposal from FAA [91]. In this case, the algorithm calculates the shortest path from the current last waypoint of the trajectory to the message’s location data, considering the airways topology (Line 7). Thus, the processed path becomes a part of the trajectory (Line 8).

On the other hand, the applied Remote ID protocol can be an enhanced version, for instance, using the *ARID* anonymization protocol [146]. In this case, the drone’s ID will change constantly. Nonetheless, our attack tries to track the trajectories even in these cases. The attacker verifies the current profiles that generate the shortest path to the m ’s location and evaluates its influence on the predicted travel velocity of the path (Lines 9–27). Hence, this step is independent of anonymization techniques.

This procedure occurs as follows. For each existent profile (Lines 13–27), the shortest path is calculated (Lines 14–15). It is possible to obtain the average velocity Δ_v to fly over the path, given both the associated time mark of the message t_M and the time mark of the last added trajectory location $\tau.t$ (Line 16). Likewise, it is possible to obtain the deviation of Δ_v from the expected average speed, used here as a reference value to define the best trajectory profile for m (Line 17).

After, the algorithm verifies two conditions. First, if the average velocity belongs

Algorithm 12: RemoteID-Tracking

```

Input  :  $m, G, T, \epsilon$ 
Output:  $T$ 
1  $t_M \leftarrow$  time mark from  $m$ 
2  $v_M \leftarrow$  drone's velocity data of  $m$ 
3  $loc_M \leftarrow$  drone's location information of  $m$ 
4  $\tau \leftarrow$  verify if  $m.ID$  is associated with a profile  $\tau \in T$ 
5 if  $\tau \neq null$  then
6    $src \leftarrow$  last waypoint of  $\tau.P$ 
7    $path \leftarrow$  shortest-path( $src, loc_M, G$ )
8   insert  $path$  at the end of  $\tau.P$ 
9 else
10   $bestMatch \leftarrow null$ 
11   $bestDeviation \leftarrow \infty$ 
12   $bestPath \leftarrow \langle \rangle$ 
13  for  $\tau \in T$  do
14     $src \leftarrow$  last waypoint of  $\tau.P$ 
15     $path, distance \leftarrow$  shortest-path( $src, loc_M, G$ )
16     $\Delta_v \leftarrow distance / (t_M - \tau.t)$ 
17     $deviation \leftarrow |\Delta_v - avg(\tau.v, v_M)|$ 
18    if ( $\Delta_v \in [\min(\tau.v, v_M) - \epsilon; \max(\tau.v, v_M) + \epsilon]$ )  $\wedge$  ( $deviation <$ 
       $bestDeviation$ ) then
19       $bestMatch \leftarrow \tau$ 
20       $bestDeviation \leftarrow deviation$ 
21       $bestPath \leftarrow path$ 
22  if  $bestMatch = null$  then
23     $bestMatch \leftarrow$  create a new trajectory profile in  $T$ 
24    assign  $m.ID, loc_M,$  and  $v_M$  to  $bestMatch$ 
25  else
26    insert  $bestPath$  at the end of  $bestMatch.P$ 
27     $bestMatch.v \leftarrow v_M$ 

```

to the interval defined by the speeds described in m and τ , added to the error ϵ . If Δ_v is outside the interval, it indicates that a drone should fly too fast or too slow from the last associated waypoint to the m 's location, not corresponding to the trajectory pattern. Second, if the deviation is less than the current best deviation (Line 18). If both conditions are true, the trajectory profile τ becomes the best candidate for m (Lines 19-21).

After verifying all the profiles, the shortest path is added to the best candidate profile (Lines 25-27). If any trajectory meets the conditions, then there is a possibility the message m was the first eavesdropped from a drone not yet known for the attack. In this case, a new trajectory profile is created and associated with m 's data (Lines 22-24).

► *Time Complexity Analysis:* Let us assume that a hashing data structure can store the IDs associated with the profiles, taking a constant time to store and access the data. The algorithm's workflow divides into the cases when the ID is known (Lines 5-8)

or when all the trajectories will be assessed (Lines 10–27).

In the first case, attribution tasks take $\mathcal{O}(1)$, and the path insertion takes $\mathcal{O}(|path|)$, considering a data structure with a pointer to its last element. However, this case is dominated by the `shortest-path` operation. For instance, if we use the Dijkstra shortest path algorithm, the time complexity is $\mathcal{O}(|E| + |V| \log |V|)$.

Regarding the second case, the operations are similar, with attribution tasks, math-based operations, and assignments following a constant time complexity. Also, the `shortest-path` function has the same time complexity as previously discussed. However, this function can be processed for all the profile trajectories in the worst case. Then, this case dominates the first one. Hence, the time complexity of Algorithm 12 is given by:

$$T(\text{RemoteID-Tracking}) = \mathcal{O}(|T||E| + |T||V| \log |V|) \quad (4.20)$$

4.6.3 Simulation Setup and Performance Evaluation

This section presents the definitions regarding our experimental evaluation. Our intention is not just to investigate to what level the Remote ID represents a threat to the IoD location privacy but also if existent privacy-based protocols allied with Remote ID can enhance this aspect. Besides the Remote ID final rule, we evaluate two other privacy-based protocols: the ARID [146], and a collaborative protocol of ARID and t-MixDrones. We consider these mechanisms for two reasons: (i) *ARID* is proposed as an enhanced solution of Remote ID, but its location privacy protection was not evaluated in its original proposal; and (ii) t-MixDrones is a LPPM for dense scenarios in IoD, presenting proper levels of location privacy. All of them are briefly described as follows.

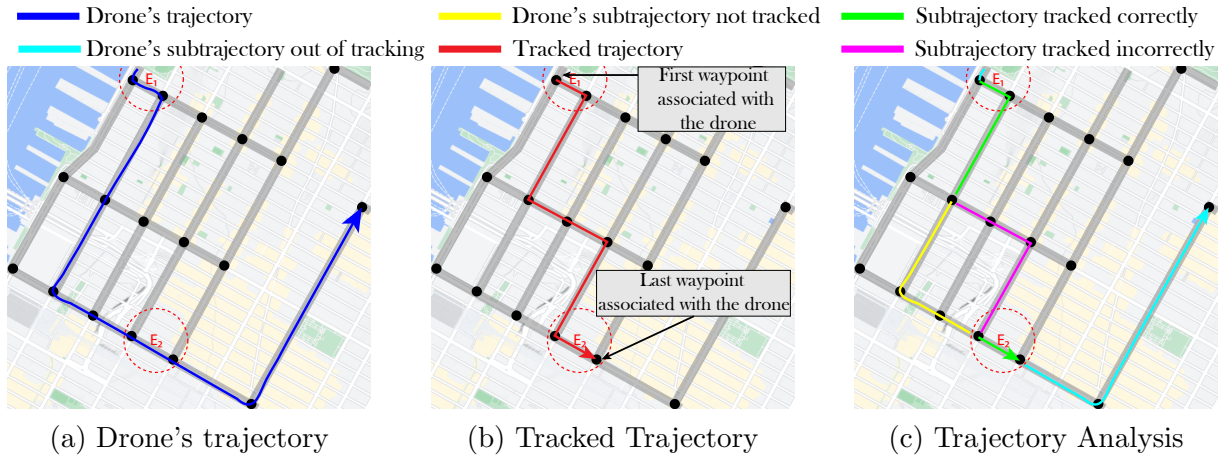
- ▶ *Remote ID*: we implement the Remote ID message following the final rule defined by FAA [91]. An application to broadcast and receive the data are available in all the drones, ZSPs, and eavesdroppers in the simulated environment;

- ▶ *ARID* [146]: we also implement a version of this protocol, generating Remote ID messages that change the drone’s identity for each broadcast message;

- ▶ *ARID* [146] + *t-MixDrones*: As discussed, t-MixDrones can enhance location privacy by changing the drone’s airway altitude as a mixing factor besides its identity. Hence, we implement a collaborative version of *ARID* and t-MixDrones, where the first manages the drone’s re-anonymization and the second “mixes” the airway altitude.

- ▶ **Metrics**: Our analysis occurs through the actual drone’s trajectory and the corresponding trajectory tracked by the attack. However, it is intuitive that the eavesdroppers

Figure 4.21: Example of trajectory analysis comparing the actual drone's trajectory and the tracked by the Remote ID-based attack



Source: Elaborated by the author

can not cover the environment entirely. Also, the tracked trajectory can not correspond with the actual drone's trajectory in its entirety, having some sub-trajectories tracked incorrectly.

Figure 4.21 shows an example of our analysis regarding a trajectory tracking analysis. This example has a top view of an excerpt of an urban scenario, where the available airspace is denoted by the dark gray regions, and the topological waypoints by the black circles. Although it seems like the trajectory is on the ground, it is at an elevated altitude. Two eavesdroppers (E_1 and E_2) are in strategic positions, denoted by the dotted red circles.

Figure 4.21a presents an example of an actual drone's trajectory, while Figure 4.21b illustrates the corresponding tracked one. As we note, the attack can track just a piece of the complete drone's trajectory due to the eavesdroppers' position coverage. Thus, we consider three different trajectories: the complete trajectory, named $CTraj$; the sub-trajectory of $CTraj$ delimited by the first and last waypoints gathered by the eavesdroppers, named $DTraj$; and the attack's tracked trajectory, named $TTraj$.

In this case, the two last waypoints do not belong to $DTraj$, representing a sub-trajectory out of tracking ($SubOT$) (light blue line in Figure 4.21c). Furthermore, the green segments match when comparing the drone, and the tracked trajectory, representing a sub-trajectory tracked correctly ($SubTC$). Pink segments, in turn, are tracked by the attack, but they are not in the actual trajectory, representing a sub-trajectory tracked incorrectly ($SubTI$). Finally, yellow segments belong to the drone's trajectory but not to the tracked circuit, meaning a sub-trajectory not tracked ($SubNT$) ($SubNT$).

Given this notation, we define the following metrics:

► **Complete Trajectory Matching Rate ($CTMR$)**: it represents the rate of

Table 4.11: Remote ID Case Study: Simulation Parameters

Parameter	Value
<i>General Parameters</i>	
Simulation time	30 minutes
Environment boundaries	$3.8 \times 10.5 \times 0.25 \text{ km}^3$
Airways Altitudes	(100, 150, 200 and 250 m)
#Drones	50
Drone speed	uniform 8–12 m/s
Mobility pattern	Gauss-Markov
# Eavesdroppers	{5, 6, 7, 8, 9, 10}
Protocols	{RemoteID, ARID, ARID+tMixD}
<i>Remote ID Parameters</i>	
Broadcast Frequency	1 sec
Communication Range	400 m
<i>t-MixDrones Parameters</i>	
#Mix Zones	4
Mix Zones coverage radius	250 m
\mathcal{P}_{change}	0.8
k -anonymity	2

$SubTC$ compared to $CTraj$. It is formally defined as:

$$CTMR = \frac{|SubTC|}{|CTraj|}$$

► **Delimited Trajectory Matching Rate ($DTMR$):** similar to $CMTR$, this metric restricts the evaluation to $DTraj$. It is formally defined as:

$$DTMR = \frac{|SubTC|}{|DTraj|}$$

► **Waypoint Segments Tracked Correctly Rate ($STCR$):** it calculates the rate of $SubTC$ in the tracked trajectory $TTraj$. This metric is defined as:

$$STCR = \frac{|SubTC|}{|TTraj|}$$

► **Waypoint Segments Tracked Incorrectly Rate ($STIR$):** this metric is similar to $STCR$ but considers the $SubTI$. This metric is defined as:

$$STIR = \frac{|SubTI|}{|TTraj|}$$

► **Application Scenario and Simulation Setup:** We conducted extensive simulations to evaluate the protocols mentioned above using the IoDSim. Table 4.11 summarizes a list of relevant parameters of our simulation. We design a topological airspace of an urban environment following the terrestrial roads of a part of Manhattan Island, NY, with

four available airway altitudes. We spread 50 drones over the environment, where they perform their flights with speeds varying between 8 and 12 m/s for 30 minutes.

These drones have well-defined path plannings with a range of scheduled trips. When a drone finishes a planned trip, it starts the next one right away. The Remote ID messages are broadcast by the drones every 1 sec with a communication range of 400 m. We vary the number of eavesdroppers from 5 to 10, spreading them strategically over the environment. For each combination of eavesdroppers \times protocol, we executed 30 simulations with different seeds, totaling 540 experiments, reinforcing our results’ robustness and statistical validation with a 95% confidence interval. These results appear in the next section.

4.6.4 Results and Discussion

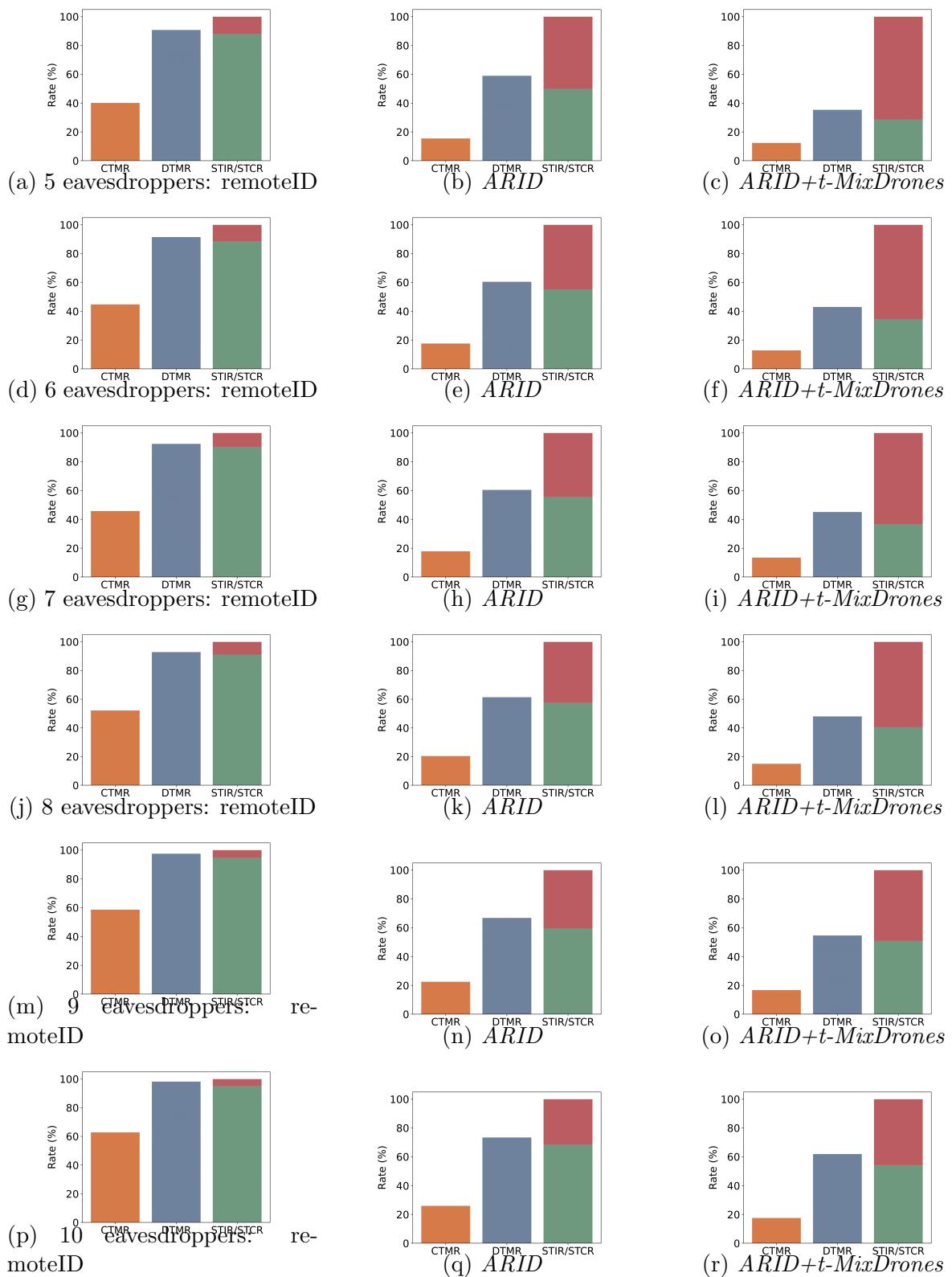
This section presents the results obtained from our extensive simulations. Figure 4.22 highlights these results, where each line corresponds to the number of eavesdroppers, and each column corresponds to a protocol. Each result represents the average of the 30 experiments related to each configuration. As the interval errors are less than 1% for all the experiments, they are not on the charts. It is important to note that the metrics are attack-centered, i.e., the greater the result, the more successful the attack, and the less the location privacy.

► **Complete Trajectory Matching Rate (*CTMR*):** The *CTMR* indicates the attack’s success in the environment, considering the entire drone’s trajectory. The orange bars represent this metric in the charts. We note that the attack has a higher *CTMR* for the Remote ID protocol. Indeed, when the Remote ID is applied, the attack can track more than the double waypoint segments compared to the other two protocols in all scenarios. With 10 eavesdroppers, the attack can track more than 60% of the trajectory, while this value remains around 20% with *ARID* and *ARID+t-MixDrones*.

We can also note that the *CTMR* increases as the number of eavesdroppers increases for Remote ID. On the other hand, this increase is much lower compared to the other two protocols. While with Remote ID the rate increases around 50% from 5 to 10 eavesdroppers, the rate has an increase of less than 5% for the others. It reveals the importance of anonymization protocols to provide location privacy. Even with more eavesdroppers, tracking remains difficult. Also, these insights indicate that Remote ID is much more susceptible to location-based attacks.

Nonetheless, these rates are far from a relevant threat regarding drone tracking,

Figure 4.22: Case study results of the evaluated metrics for each combination of number of eavesdroppers \times protocol



Source: Elaborated by the author

leading to other attacks, such as hijacking. It can be directly related to the insufficient

number of eavesdroppers over the environment. Indeed, the experiments have one eavesdropper per 3.9 km², considering the max number of them.

Concerning the message's broadcast communication range of 400 m, the airspace is not entirely covered. Also, the eavesdropper's position can affect the attack performance. All in all, these aspects must be further investigated.

► **Delimited Trajectory Matching Rate (DTMR):** *DTMR* provides an accurate overview of the attack's success since it compares the tracked trajectory to the drone's trajectory delimited by the first and last waypoints gathered by the eavesdroppers. This metric is represented by the blue bar in the charts.

Likewise, in *CTMR*, the attack has an outstanding success facing the Remote ID. For all eavesdroppers, the attack can track more than 90% of the drone's trajectory, while the performance for *ARID* and *ARID+t-MixDrones* varies between 60–75% and 38–60%, respectively. These aspects reinforce that Remote ID can not provide a proper level of location privacy in which a considerable portion of its trajectory can be inferred by malicious entities equipped with a Remote ID module.

Analyzing *CTMR* and *DTMR* together, we can note a significant increase mainly for Remote ID and *ARID*. However, when *ARID* and *MixDrones* are cooperating, the attack has constrained success, mainly for a small number of eavesdroppers. This enhanced protection can be related to the change of the drone's altitude performed by *t-MixDrones*. As there are few eavesdroppers, processing the shortest path between the gathered waypoints involves a greater distance, which is difficult to infer correctly where an altitude change occurred.

► **Waypoint Segments Tracked Correctly Rate (STCR), and Waypoint Segments Tracked Incorrectly Rate (STIR):** We discuss the results of both *STCR* and *STIR* together as they are complementary metrics. The green bars represent *STCR*, while the red bars represent *STIR* rates. Likewise, in the former two metrics, the attack has a higher *STCR* with Remote ID when compared to the other two mechanisms. For all the experiments with Remote ID, the attack generates a trajectory with more than 90% of waypoint segments equal to the delimited drone's trajectory. Analyzing the generated trajectories thoroughly, we can note that incorrect waypoint segments are generated mainly when the possible paths, given a source and destination waypoints, have a similar distance or when these waypoints are considerably distant, increasing the possible paths.

Also, we can observe that the *STCR* follows a similar rate to the *DTMR* of the related experiment. This relation shows a similarity between the trajectories, where the number of incorrect waypoint segments is close to the segments not tracked (considering the delimited drone's trajectory). Nonetheless, this relation of similarity demands further research, where we can apply metrics of trajectory similarity.

► **General Discussion:** This experimental evaluation demonstrated that Remote ID is a threat to the drone’s location privacy. Undoubtedly, the accurate tracking of more than 90% of the drone’s delimited trajectory (considering the one delimited by the first and last waypoints gathered by the eavesdroppers) is a high-risk factor for location privacy in an IoD environment. From the low to a high number of eavesdroppers, the success of the applied attack is always evident. Even considering that the attack cannot track a substantial part of the drone’s complete trajectory (as demonstrated by the *CTMR* metric), there is no guarantee of protection from the Remote ID since there is a direct dependence on the strategic positioning of the eavesdroppers.

We also demonstrate that existent LPPMs can enhance the location privacy level of the IoD when treated as an extension of the standard Remote ID. In this case study, ARID [146] and t-MixDrones proved to protect the drone’s identity and location information. They considerably hamper the attack’s success through successive anonymizations of the UA ID field in the Remote ID’s message and in-flight maneuvers, such as mixing the drone’s altitude.

In a nutshell, we strongly affirm that Remote ID needs the implementation of strategies that value the privacy of your information, such as location. As technically demonstrated in this study, it is possible to achieve this requirement using existing LPPMs without harming the FAA’s initial objectives in airspace monitoring.

4.7 Chapter Remarks

This chapter presented our contributions in the research field of Location Privacy Protection Mechanisms for the IoD. We designed three novel LPPMs, namely, t-MixDrones, MixRide, and TDG. The two first mechanisms are based on the Mix Zones concept. The performance evaluation pointed out that they can ensure suitable levels of location privacy for environments with a dense number of drones when compared to traditional MZ-based mechanisms. However, we noted that t-MixDrones increased the drone’s battery consumption, which justifies the design of MixRide since this latter is an energy-aware approach. TDG, in turn, is designed for sparse environments, regardless of the number of drones. We demonstrated that the presence of airways, inherent to IoD, makes the existing dummy-based approaches ineffective in this environment, justifying the designing of TDG. The proposed mechanism also addressed suitable levels of location privacy, preserving both the identity and localization of drones when shared with 3rd-parties services.

Therefore, we are able to answer the research questions of this dissertation consid-

ering the design of LPPMs for IoD:

► **Can the existing Location Privacy Protection Mechanisms provide the same protection level to IoD environments when compared to traditional mobile networks?**

Answer: Considering MZ and dummy-based LPPMs, they can not. As discussed in this chapter, the particular characteristics of the IoD environment pose several shortcomings to the application of these existing mechanisms in such a way that the provided location privacy can not ensure a suitable security level for drones.

► **Is it possible to adapt these existing LPPMs to enhance the protection level provided to a given IoD environment?**

Answer: Yes, it is possible. Through extensive performance evaluation, the three proposed LPPMs demonstrated they can provide a suitable level of location privacy for IoD, considering the specific environment conditions they are related.

We also proposed in this chapter the IoDAPM, an RL-based strategy to assign dynamic the proposed LPPMs in the IoD, given the heterogeneous conditions observed in the environment. IoDAPM is a robust and useful approach to deploy in IoD since there is no “silver bullet” regarding the LPPM coverage for IoD, where each one has its pros and cons. Through an extensive performance evaluation, we demonstrate that IoDAPM can extract the best of their advantages, assigning the best one periodically, enhancing the provided QoS for the related services.

Furthermore, we presented a case study regarding the importance of location privacy in the IoD context, and how the existing governmental policies must be in accordance with these principles. We technically demonstrated through the design of a location-based attack that the Remote ID rule, proposed by the FAA, is a threat to the drone’s location privacy considering its final proposal. To tackle this issue, we proposed an enhanced design of Remote ID, incorporating different privacy-related mechanisms in its model.

Last but not least, we identified some challenges related to this research field. Summarily, there is a critical trade-off between the energy efficiency and the associated trip delay of drones, which can affect the provided QoS deeply. This challenge is highlighted by t-MixDrones and MixRide, mainly by the latter in its configuration tuning analysis. Furthermore, the TDG performance evaluation revealed a shortcoming regarding dummy-based mechanisms for airway-oriented IoD environments: the actual location of the drone, while preserved, does not cause a significant disruption to an adversary due to geographic airway limitations. Although IoDAPM is able to mitigate these issues, new approaches shall be investigated to overcome these challenges.

Chapter 5

Design of Anti-Jamming Mechanisms for IoD

This chapter presents our contributions regarding the design of anti-jamming mechanisms for IoD. From a discussion about how restricted flyable airspace affects UAVs path planning, we design an Anti-Jamming mechanism, named IoD-JAPM, to mitigate the effects of JA in the IoD environment. The design of this mechanism is also guided by the proposed framework to overcome the performance of the existing approaches, being applied in different situations.

This chapter is organized as follows. Section 5.1 brings an introduction and motivation discussing why current strategies do not meet the IoD characteristics, mainly the flyable space constrained by airways. Section 5.2 presents the application scenario and threat model. Next, Section 5.3 discusses the related studies, highlighting how IoD-JAPM can overcome the opened challenges. The formal modeling of the proposed mechanism is presented in Section 5.4. In Section 5.5, we describe the simulation setup and performance evaluation. We analyze the obtained results in Section 5.6. Lastly, Section 5.7 brings the chapter remarks.

5.1 Introduction

In the former chapters, we discuss how location-based attacks can affect the IoD environment, mainly regarding the drone's location privacy. However, other attacks can affect directly this privacy aspect without discovering the drone's location through the analysis of a group of queries. One of these attacks is Jamming Attack (JA), introduced in Section 3.1.4. JA represents a severe risk in IoD because availability is a paramount requirement in this network. There are some effective countermeasures against JA in terrestrial mobile networks (discussed in Section 3.2.3). However, they are ineffective in IoD because they can not appropriately handle LoS-induced security issues [51].

Once a drone has its availability affected, it can present an unusual behavior, such as landing at inappropriate places, leading to other attacks, such as hijacking. In some situations, an airway can have all its flyable airspace affected by a JA, making the drones that fly over there incommunicable [88, 150]. These issues can also affect the trajectory of drones, leading to a revised path planning.

The relation between JA and the drone trajectory has been widely discussed in UAV-based networks [44, 51, 88, 89, 90, 150, 151, 152, 153], but those studies consider the airspace free to fly. In contrast, one of the most prominent characteristics of IoD is the presence of airways, allowing drones to fly over constrained airspace. Recent studies consider IoD organizing airspace through airways, demonstrating their importance for adequately managing the flight of drones [154, 155]. Since the airways limit this flyable airspace, current solutions can not be applied directly to IoD.

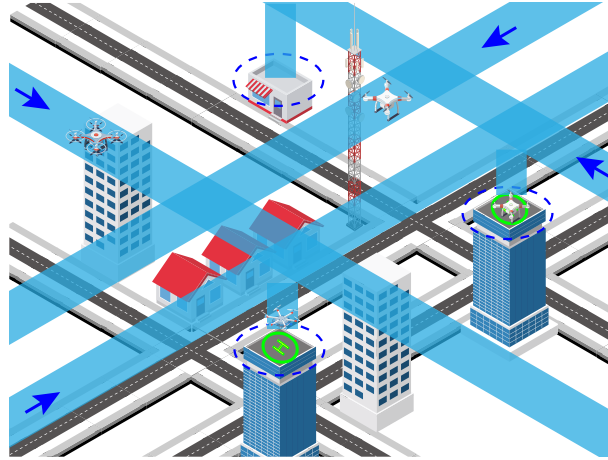
Bearing these challenges in mind, we investigate the impact of JA on drone path planning and, therefore, its trajectory. From that, and considering the proposed framework to guide the design of PMs, we designed the IoD-JAPM, an airway-aware protection strategy against JA on the IoD. The mechanism ranges from analyzing the airway's availability to reformulating the drone path planning. IoD-JAPM embraces a method to isolate a region affected by a JA in the IoD, considering the airway topology; a strategy to avoid a jamming signal in an affected region without violating the flight restrictions imposed by the airway; and another strategy to mitigate the impact of the reformulated drone path planning when its final destination belongs to a region affected by the JA. This strategy considers the presence of *vertiports* in the environment.

5.2 Application Scenario and Threat Model

Figure 5.1 illustrates the IoD environment as an envisioned ITS scenario, where the blue segments are the flyable airways. As we can note, they are parallel to the grounded roads. Furthermore, these environment has PoIs, such as *vertiports* and delivery pick-up regions (represented by the dashed blue circles). Drones use dedicated airways to access these PoIs. As discussed, the topology can be modeled as a graph G in which, given two intersection nodes, a directional edge indicates how a drone can fly between the nodes.

Regarding the threat model, we consider that a set of attackers \mathcal{A} can perform a JA as a denial of service, affecting the nodes' availability in the environment. The attack occurs when adversaries interfere in the communication between a set of network nodes by flooding the network communication channels, occupying a part of (or the whole) bandwidth. Hence, the nodes can not use the network because the communication channel is

Figure 5.1: IoD environment as an ITS



Source: Elaborated by the author

always occupied. Multiple stationary adversaries can be distributed over the environment. They will perform the attack considering the best configuration for them.

A given adversary can perform short-term and long-term attacks. An attacker also can not perform a JA. Thus, given an attacker $\alpha \in \mathcal{A}$ and a time interval Δ_t whose initial time is t_0 and the last time is t_n , these attacks can be characterized as the following:

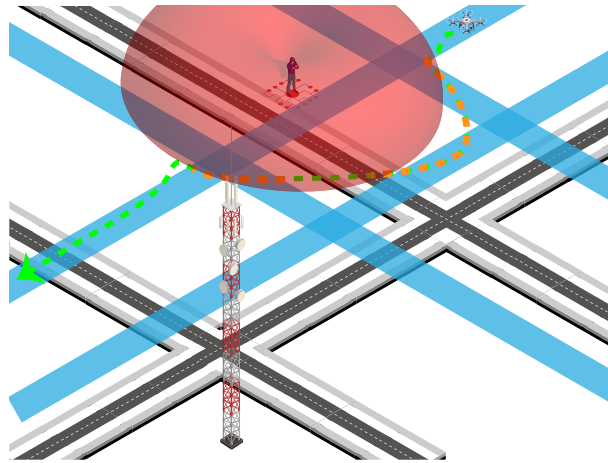
- *Absence of JA*: $\sum_{k=0}^n J_k = 0$;
- *Long-term JA*: $\forall k \in \langle 0, n \rangle : J_k \neq 0$, meaning the attack remains over all Δ_t ;
- *Short-term JA*: given a time threshold t_i , $\sum_{k=0}^i J_k = 0$ and $\sum_{k=i+1}^n J_k \neq 0$, or vice-versa. It means, the attack is performed in a short period of time. The subsequent occurrence of this attack indicates the adversary is performing intermittent JA.

5.3 Related Studies

In recent years, JA has been investigated in the context of the UAV-based network as a threat and security mechanism [156]. Specifically, some studies focused on analyzing how JA interferes with the drone trajectory. The majority of these proposals consider a grounded and stationary jammer.

Wang et al. [151] discussed how JA affects the UAV's trajectory, investigating a scenario where the UAVs communicate with grounded nodes suffering a jamming signal from a stationary source. They formulated this challenge as an optimization problem, designing

Figure 5.2: Example of how JA affect drone trajectory in IoD



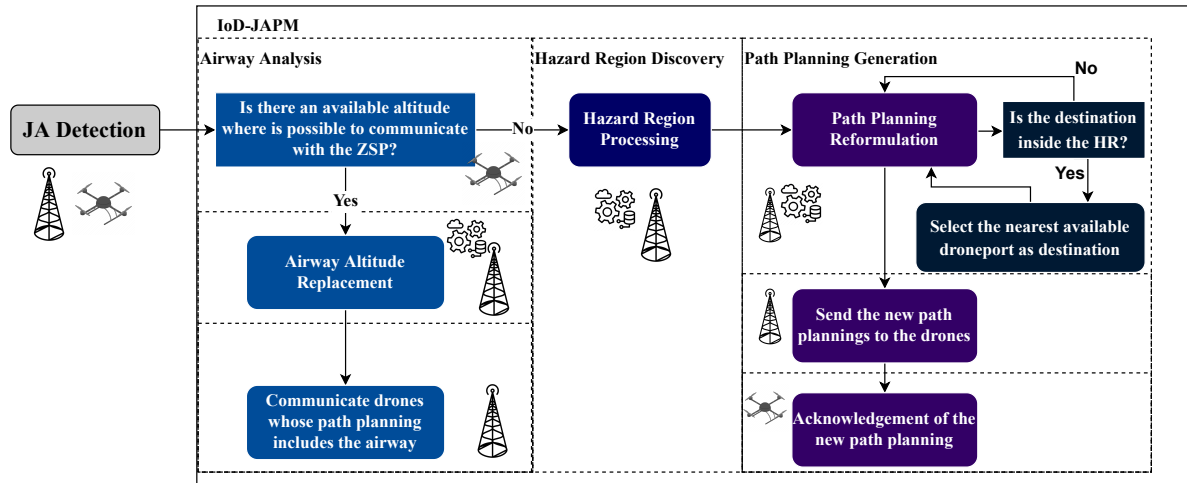
Source: Elaborated by the author

a trajectory planning method to optimize the drone's position in the airspace dynamically, applying the concept of slack variables. Although they addressed optimal results for a single adversary, the solution generates sub-optimal results for multiple jammers. Wu et al. [51, 152] expanded this context by exploring the QoS involved in considering the communication throughput and delay with grounded nodes, whose location guides the UAV trajectory. They proposed two-block coordinate descent (BCD) based algorithms to solve it sub-optimally, which improved significantly different QoS requirement levels. Gao et al. [44] enhanced this model by using secondary transmitters. Duo et al. [150] proposed a similar solution but considered a scenario where drones collect data from a Wireless Sensor Network (WSN).

Some studies investigated the impact of JA in other networks integrating drones as a relay communication node, such as VANETs [89, 153]. They model an anti-jamming game in which the UAV decides whether or not to relay the grounded vehicle's message to a Road Side Unit (RSU). Recently, intelligent anti-jamming mechanisms for UAV-based networks considered machine learning-based strategies. Sedjelmaci et al. [90] proposed a reinforcement learning-based approach that handles cyber-attacks, including JA. Mowla et al. [88] designed a distributed mechanism based on Federated and Reinforcement Learning for FANETs, discussing that a distributed approach can cover several challenges of UAV-based networks, for instance, the Size, Weight, and Power (SWaP) limitations.

It is noteworthy that JA has been widely investigated in the UAV context. On the one hand, all these studies assume that drones can fly freely over airspace. On the other hand, drones have limited airspace to fly in IoD. Given that a particular region is affected by JA, a drone flying over there will be under attack. Considering the airway boundaries, the drone may not have a possible trajectory to avoid the attack regardless of the strategy adopted. In a previous study [157], we investigated these challenges, providing a baseline anti-jamming mechanism for IoD that reformulate the drone's path planning

Figure 5.3: The proposed IoD-JAPM workflow



Source: Elaborated by the author

aiming to avoid aerial regions affected by JA. In a nutshell, the proposed mechanism identifies and isolates a region affected by JA, reformulating the path planning of drones. Although it provided adequate protection to the drones against JA, it also revealed that constrained airway topologies tend to hamper the provided protection. Moreover, whenever the number of jammers increases, there is an increment in the flight distance. Also, the mechanism can not handle situations when the drone destination belongs to a region suffering an active JA. All these issues represent challenges to investigate.

Fig. 5.2 illustrates an example of a grounded adversary performing a JA over the IoD. The blue segments represent the airways, the red region represents the range of the performed JA, and the dashed line represents the drone trajectory. Considering the current solutions, for instance, the trajectory optimization proposed by Wu et al. [51], the drone trajectory will deviate, aiming to optimize its transmit power and the minimum delay considering the initial path planning. As we note, this deviation (orange dashed lines) can lead the drone to invade other airways not initially in its flight plan. Also, the deviation can lead to extra power consumption. These aspects pose severe risks to the availability and integrity of other drones since both collisions and power issues can occur, representing open challenges.

Given this perspective, we design IoD-JAPM, an enhanced mechanism to mitigate the JA in the IoD. We present the mechanism design in the next section.

5.4 Design of IoD-JAPM

Fig. 5.3 illustrates the main steps of IoD-JAPM. Our approach is a distributed system in which drones and ZSPs operate cooperatively to avoid an ongoing JA. This attack was previously identified by some well-known jamming detection techniques [158]. As these detection techniques perform properly for UAV-based networks regardless of the airways topology, IoD-JAPM is not limited nor dependent on these methods.

IoD-JAPM has three main steps:

► **Airway Analysis:** in the first step, a drone in the region affected by a JA stops and elevates its altitude aiming to find a re-connection with a ZSP. If the drone can re-establish communication, there is an available altitude at which JA can be avoided. This approach is based on the available aerial space between two parallel airways and the LoS communication between the ZSPs and the drones, which are particular characteristics of the IoD. We formally describe this step in Section 5.4.1.

► **Hazard Region (HR) Discovery:** if the communication can not be re-established, we move to the second step, where the IoD cloud system must discover all the affected areas, representing the Hazard Region. We formally describe this step in Section 5.4.2.

► **Path Planning Generation:** Once the HR is discovered, the IoD cloud system generates novel path plannings for the affected drones, avoiding the HR. However, this reformulation can not be addressed when the drone destination is inside the HR. In this case, IoD-JAPM selects the nearest available *vertiport* as a new destination, providing a “waiting area” for the affected drone. We formally describe this step in Section 5.4.3.

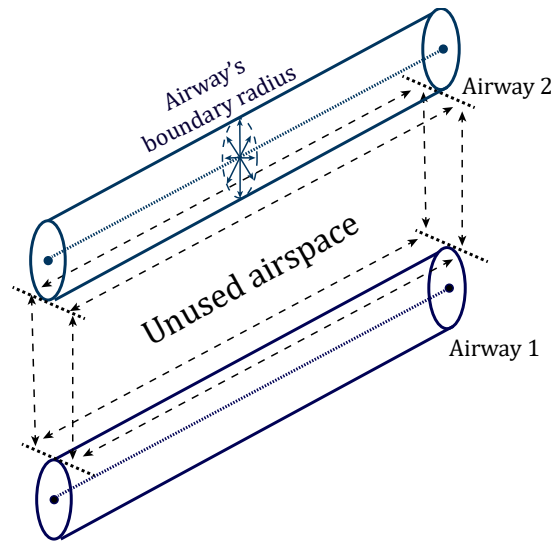
Before defining the IoD-JAPM in detail, we formalize the main elements involved in this design (beyond the elements previously defined). They are described as follows.

- \mathcal{PP} is the set of path plannings defined for each $d \in \mathcal{D}$;
- \mathcal{V} is the set of *vertiports* over the environment;
- τ is an interference signal over the communication channel, representing the JA properly. This signal stems from a subset $\mathcal{A}' \subset \mathcal{A}$.

As discussed, we propose IoD-JAPM to overcome the lack of current solutions for the IoD environment, focusing on the drone trajectory through the airways. Therefore, our focus is to design a mechanism that allows the network to recover promptly when an attack is previously detected. Thus, we take the following these assumptions:

- IoD nodes can detect a JA’s occurrence following some detection strategies found in the literature [88, 159, 160]. Our mechanism is not limited nor dependent on these

Figure 5.4: Delimitation of unused aerial space between two parallel airways with different altitudes



Source: Elaborated by the author

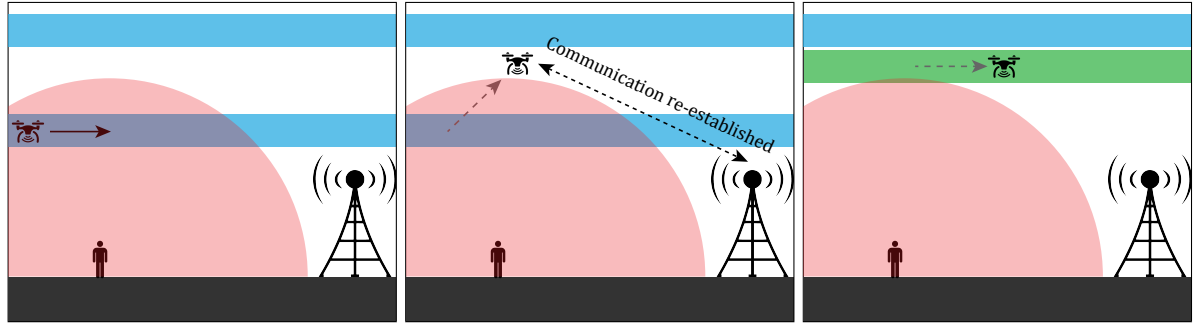
detection strategies, being applied one step forward when the attack is already detected;

- A given adversary will always perform the attack through the best configuration, considering his goals. The adversary can perform a continuous JA regardless of the power consumption of the jamming device. Furthermore, the JA can be driven either to affect the communication between the nodes or to the GPS signals;
- Once a given drone is affected by the attack, it can lose communication with the ZSP. In this case, the drone generally hovers or returns to the last point of a valid communication. If any anti-jamming strategy is taken, the network facilitates other attacks, such as purposeful hijacking;

5.4.1 Airway Analysis

As presented in Fig. 5.3, once the network detects the occurrence of a JA, the IoD-JAPM first step consists of a drone verifying if it is possible to establish communication with a ZSP node z still inside the airway in the affected region. Considering that in our system model, the attack occurs from the ground, τ strength decreases as the drone altitude increases.

Figure 5.5: Airway analysis step of IoD-JAPM performed by a drone



(a) A drone inside an affected region loses its communication with the ZSP
 (b) The drone elevates its altitude trying to re-establish the communication
 (c) The ZSP set a new altitude to the affected airway

Source: Elaborated by the author

Hence, the main idea in this step is to move the drone to a higher altitude aiming to re-establish communication with a ZSP z . However, the airways have a constrained region to flight, including a maximum altitude. Nonetheless, once an airway segment β is placed, other segments should not intersect β in lower or higher altitudes unless through the insertion of an intersection node [5]. An exception occurs when β has similar airway segments defined through different altitudes, as depicted in Figure 5.4. Given these assumptions, a drone d flying over an affected airway segment β has a safe altitude range to fly, ranging from its current position to a maximum altitude α . This maximum altitude can be previously defined by the ZSP when a drone requests access to the airspace, considering two main aspects: a safe distance from a parallel high-altitude airway segment when it exists; and the drone SWaP constraints. Hence, the drone can try to re-establish the connection with the ZSP over unused airspace, as illustrated in Figure 5.4.

Formally, the airway analysis step embraces both the drone and ZSP systems. Hence, we define two algorithms for this step. Algorithm 13 is a drone-centered method. Algorithm 14, in turn, is the ZSP-centered procedure.

► **Drone-centered Approach:** Algorithm 13 formally describes the airway analysis carried out by a given drone. Figure 5.5 illustrates this central concept. As input, it requires the pre-defined maximum altitude α , the information of the ZSP z , and the time intervals Δt_{ack} and $\Delta t_{schedule}$, representing an acknowledge and schedule time intervals, respectively.

Initially, the drone tries to communicate with z , sending a **Hello** message, verifying whether the communication still works (Line 1). If, after a time interval Δt_{ack} , there is no **ACK** from z , the drone assumes that the communication is lost, initializing the communication attempts on higher altitudes (Lines 2-9), as illustrated in Figure 5.5a. Based on the current drone altitude (Line 3), it adjusts its trajectory planning if its current altitude does not reach α (Line 5). After, the drone schedules a new communication attempt

Algorithm 13: Drone-Airway-Analysis

Input : $\alpha, z, \Delta t_{ack}, \Delta t_{schedule}$

- 1 send a **Hello** message to z and wait a time interval Δt_{ack} for an **ACK** answer
- 2 **if** *There is no ACK from z* **then**
- 3 $\alpha_{current} \leftarrow$ current drone altitude
- 4 **if** $\alpha_{current} \leq \alpha$ **then**
- 5 Adjust the navigation vectors towards α
- 6 $t_{next} \leftarrow \Delta t_{schedule} +$ the current system time
- 7 schedule a call to **Drone-Airway-Analysis** on t_{next}
- 8 **else**
- 9 go back to the previous checkpoint // All attempts to communicate fail
- 10 **else**
- 11 send a **Free-JA** message to z informing $\alpha_{current}$
- 12 Adjust back the navigation vectors toward the current checkpoint

based on the time interval $\Delta t_{schedule}$ (Line 6). Suppose the current altitude reaches α (Lines 8-9). In that case, all attempts fail. The ZSP will need to take additional action, isolating the affected region and reformulating the drone path planning, as described in the following subsection. In this case, the drone returns to the last checkpoint where communication occurred.

However, suppose the drone receives the **ACK** from z (Lines 11-12). In that case, the communication is re-established, indicating that a given drone flying over the current altitude $\alpha_{current}$ is not affected by the JA, as demonstrated in Figure 5.5b. In this case, the drone sends a **Free-JA** message to z , informing the current altitude and adjusting its trajectory to the present checkpoint according to its path planning.

Since z receives the new altitude, it will update this information regarding the affected airway segment and spread it to the drones whose path planning contains the affected segment. Figure 5.5c shows this case, in which the airway with green color represents the affected airway with the new altitude. As these procedures consist of update-based messages, they are not described in detail throughout the algorithms.

► *Time Complexity Analysis:* in this analysis, we focus on the time complexity of the procedures inherent to the drone system. Therefore, this analysis does not cover “inter-node” procedures (Lines 1 and 11). In summary, Algorithm 13 consists of direct assignments to primitive data structures inherent to the drone system (Lines 3, 5, 6, 7, 9, 12). Hence, the time complexity of Algorithm 13 is given by Equation 5.1:

$$T(\text{Drone-Airway-Analysis}) = \Theta(1) \quad (5.1)$$

► **ZSP-Centered Approach:** Bearing that drones can not re-establish communication with the ZSP even after analyzing other altitudes, a given ZSP z monitors the drone location update. Thus, the IoD cloud system can take other actions to avoid the JA (defined in the following sections). Algorithm 14 describes the ZSP monitoring. Besides

Algorithm 14: ZSP-Check-Drone-Upd

Input : $\alpha, \mathcal{Q}_{upd}, \mathcal{D}_{JA}$

- 1 $d_{upd} \leftarrow$ extract element of the top from \mathcal{Q}_{upd}
- 2 **if** d_{upd} did not update its location when expected **then**
- 3 **if** $d_{upd} \notin \mathcal{D}_{JA}$ **then**
- 4 $\mathcal{D}_{JA} \leftarrow \mathcal{D}_{JA} \cup d_{upd}$
- 5 $l_{last} \leftarrow$ get last location update of d_{upd}
- 6 $t \leftarrow$ predict the time that d_{upd} will reach α from l_{last}
- 7 insert d_{upd} in \mathcal{Q}_{upd} with priority t
- 8 schedule a call to ZSP-Verify-Drone-Update at t
- 9 **else**
- 10 $l_{predict} \leftarrow$ predict the d_{upd} 's location if it had updated its location
- 11 Cloud-Calc-Hazard-Region($l_{predict}$)

the maximum altitude α , the algorithm requires as input a priority queue \mathcal{Q}_{upd} and a set \mathcal{D}_{JA} . \mathcal{Q}_{upd} stores the information of drones ordered by the next expected location update time. \mathcal{D}_{JA} stores drones trying to find an altitude free from JA. The recurring drones' location update messages fed the data structures inherent to the IoD management system.

Initially, the mechanism takes d_{upd} (that would have updated its position) from \mathcal{Q}_{upd} (Line 1). If d_{upd} did not update its location at the current time, it could indicate a possible communication loss due to the JA (Lines 2-11). However, it is necessary to consider the time interval the drone potentially takes to reach α . Hence, the algorithm verifies if d_{upd} already belongs to \mathcal{D}_{JA} (Line 3). The negative case corresponds to the first time d_{upd} has not updated its location. Therefore, the mechanism includes it in \mathcal{D}_{JA} (Line 4) and predicts the expended time t to reach the altitude α , based on the d_{upd} 's last location update (Lines 5-6). The d_{upd} is re-inserted in \mathcal{Q}_{upd} , now with the priority t (Line 7). Finally, the mechanism schedules a call to this procedure at the time t , aiming to verify if a new altitude was discovered during this time (Line 8).

In the affirmative case, d_{upd} has failed to find an available altitude to re-establish the communication, in which the IoD cloud system must take additional countermeasures to avoid JA (Lines 10-11), as described in Fig. 5.3. Thus, the mechanism predicts the location of d_{upd} if it had updated its position, and passes this information as input to Algorithm 15, responsible for identifying the Hazard Region (HR), leading us to the next step of IoD-JAPM.

► *Time Complexity Analysis:* to conduct this analysis, we consider that the element of the top of \mathcal{Q}_{upd} did not update its location when expected, representing the worst case. Also, let us consider a heap data structure for \mathcal{Q}_{upd} . The extraction of the top element takes $\mathcal{O}(\log n)$ (Line 1). Considering that the drone d_{upd} does not belong to \mathcal{D}_{JA} (Lines 3 to 8), both the set union and the l_{last} assignments have a constant time complexity $\mathcal{O}(1)$

(Lines 4 and 5). On the other hand, the priority queue insertion of Line 7 takes $\mathcal{O}(\log n)$, dominating this conditional. If the drone already belongs to \mathcal{D}_{JA} , the drone location prediction (Line 10) and the calling for Algorithm 15 also take $\mathcal{O}(1)$. Hence, the time complexity of Algorithm 14 is given by Equation 5.2:

$$\begin{aligned} T(\text{ZSP-Check-Drone-Upd}) &= 2\mathcal{O}(\log n) + \mathcal{O}(1) \\ &= \mathcal{O}(\log n) \end{aligned} \tag{5.2}$$

5.4.2 Hazard Region

If the drones fail to identify an altitude free from JA, the next step consists of isolating the affected region, named *Hazard Region* (HR). Considering the defined graph representation, a given location l is part of an airway line segment delimited by $\langle w_1, w_2 \rangle \in G.E$, such that the direction vector is $w_1\vec{w}_2$. Hence, to delimit the HR, the system considers three different cases, described as follows.

1. The $\langle w_1, w_2 \rangle \in G.E$ that l belongs must be part of HR;
2. $\forall \langle u_1, u_2 \rangle \in G.E$, if $\langle u_1, u_2 \rangle$ is inside a spherical HR with radius r from the l 's geographic coordinates, then it must be part of HR;
3. $\forall \langle u_1, u_2 \rangle \in \text{HR}$, $\forall \langle v, u_1 \rangle \in G.E$, if the node v has u_1 as the unique adjacent node, then $\langle v, u_1 \rangle$ must be part of HR. In other words, airway segments that reach an already affected segment and have no other airway segment to follow also must be part of HR;

Algorithm 15 defines these cases. It gives as output a subgraph $G_{HR} \subset G$ that represents the HR. Initially, G_{HR} is initialized empty (Line 1). After, all G edges are visited (Lines 2-6) to verify whether the airway segment is close enough to l considering the radius r . In the affirmative case, the segment is included in the HR (Lines 4-6). As the distance between l and its airway is near 0 (and less than r), this step considers cases (1) and (2). With the initial HR established, the system focuses on verifying if there are remained airway segments that have no other airway segment to follow - case (3). Thus, we define a subgraph $G_{rmnd} \subset G$ whose edges do not already belong to HR (Line 7). If there is an airway segment $\langle u, v \rangle \in G_{rmnd}$ that goes to the HR with any other airway to follow, it reaches a node $v \in G_{rmnd}.V$ whose outdegree is 0. It is always true since the affected airway segments are in G_{HR} .

Algorithm 15: Cloud-Calc-HR

Input : l
Output: G_{HR}

- 1 $G_{HR} \leftarrow \emptyset$
- 2 **foreach** $\langle w_1, w_2 \rangle \in G.E$ **do**
- 3 $dist \leftarrow$ calculate the distance from l to the line segment $\langle w_1, w_2 \rangle$
- 4 **if** $dist \leq r$ **then**
- 5 $G_{HR}.V \leftarrow G_{HR}.V \cup \{w_1, w_2\}$
- 6 $G_{HR}.E \leftarrow G_{HR}.E \cup \{\langle w_1, w_2 \rangle\}$
- 7 $G_{rmnd} \leftarrow G - G_{HR}$
- 8 **while** $(\exists v \in G_{rmnd}.V, outdegree(v) = 0) \wedge (v \in G_{HR})$ **do**
- 9 **while** $\exists \langle u, v \rangle \in G_{rmnd}.E$ **do**
- 10 $G_{HR}.E \leftarrow G_{HR}.E \cup \{\langle u, v \rangle\}$
- 11 $G_{rmnd}.E \leftarrow G_{rmnd}.E - \{\langle u, v \rangle\}$
- 12 $G_{HR}.V \leftarrow G_{HR}.V \cup \{v\}$
- 13 $G_{rmnd}.V \leftarrow G_{rmnd}.V - \{v\}$
- 14 Cloud-Reformulate-PP(G_{HR})

Another case can occur when the IoD airway topology has sink nodes that represent, for instance, a drone company garage. To avoid this case, it is necessary to verify if the node $v \in G_{HR}$ (Lines 8-13). When these conditions are satisfied, every airway segment that goes to v is added to G_{HR} and removed from G_{rmnd} (Lines 9-11). At the end of this iteration, the incidence list of v will be empty. Thus, v is added to G_{HR} and removed from G_{rmnd} (Lines 12-13). When there are no more nodes with outdegree equals 0, the HR is completely discovered and can be used to identify the drones needing a path planning reformulation (Line 14).

► *Time Complexity Analysis:* the initialization of the HR as an empty set (Line 1) has $\Theta(1)$ complexity. The loop from Line 2 to 7 iterates over all edges of G , performing constant mathematical operations (Line 3) and set operations (Lines 5 and 6). All of them have $\mathcal{O}(1)$ complexity. Hence, the loop is $\Theta(|G.E|)$. The deployment of the remained graph G_{rmnd} occurs through set operations (Line 7), i.e., it is $\Theta(1)$.

The remained part of the algorithm consists of two nested loops (Lines 8 to 13). Considering an aggregate analysis, these loops will iterate over the edges, in the worst case. Hence, the complexity is $\mathcal{O}(|G.E|)$. Lastly, a call to Algorithm 16 is described in the next section. Equation 5.3 gives the time complexity of Algorithm 15:

$$\begin{aligned}
 T(\text{Cloud-Calc-HR}) &= \Theta(1) + \Theta(|G.E|) + \mathcal{O}(|G.E|) \\
 &= \Theta(|G.E|)
 \end{aligned} \tag{5.3}$$

5.4.3 Path Planning Reformulation

Algorithm 16 describes the reformulation of the path planning performed in the IoD cloud system. It consists of identifying if, given a path planning $p \in \mathcal{P}$, there are airways into the HR that the drone $p.d$ will fly until the end of its trip. In the affirmative case, the system generates a new one and communicates it to the drone. Furthermore, if the drone destination point is not reachable anymore, the system tries to redirect the destination to a *vertiport* until the jamming source is actively countered.

As input, it receives the HR graph representation provided by Algorithm 15. To reformulate the path planning, we calculate first what airways are not affected by the HR (Line 1). Then, the mechanism verifies each path planning $p \in \mathcal{P}$ to identify if it will demand a reformulation. There are two auxiliary nodes, u and v , representing the last geographic point before the current path enters HR and the first geographic point after leaving HR. They are both initialized as null pointers (Line 3). Also, we initialize as *false* a flag regarding the need to designate a *vertiport* as a destination point to the drone, in the case it is not possible to generate a new path (Line 4).

Similarly, an auxiliary path p_{aux} records the reformulated path planning. It is initialized with the same attributes as p except the path planning, which is given until the current airway that the drone is flying, indicated by the *order* attribute (Line 5). After, IoD-JAPM verifies each airway segment $e \in p.G'.E'$ from the current drone airway to the last one (Lines 6-22). If the verified e is in the HR (Lines 7-9), it is necessary to check if it is the first segment inside the HR, which means that u points to null. In this case, u receives the origin node of e (Lines 8-9).

When e does not belong to HR (Lines 10-22), we need to consider two different situations regarding the previous edges: they are outside the HR, which means that e follows a “free HR path”; or part of them is in the HR. In the first case, e is added to the path in p_{aux} (Lines 11–12). Otherwise, the origin point $e.w_1$ is the first point outside HR associated with v (Line 14). Therefore, given points u and v , the minimum path planning over G_{avail} is calculated (Line 15). If there is an available path E_{new} , it is added to the new course (Lines 16-18). As an HR segment is processed and replaced, both u and v are set to null (Line 19). However, if the algorithm does not return any path, there is no alternative path to the drone. Thus, the system must process if there is an available *vertiport* where the drone can hold on until the jamming source is countered actively (Lines 20-21). After processing the airway segments of p , the system must certify if the last segment is not inside the HR. It can be verified through u , which must be null. If the segment is inside the HR (Lines 23-24), there is no alternative path to finish the trip. Likewise, in the case of Lines 20-21, it is necessary to evaluate if there is an available *vertiport*.

Algorithm 16: Cloud-Rfrmlt-PP

```

Input :  $G_{HR}$ 
1  $G_{avail} \leftarrow G - G_{HR}$ 
2 foreach  $p \in \mathcal{P}$  do
3    $u, v \leftarrow null$ 
4    $needsVertiport \leftarrow false$ 
5    $p_{aux} \leftarrow \langle p.d, p.G'.E'_{(0..p.airway.order)}, p.airway \rangle$ 
6   for  $e \in p.G'.E'$  such that  $p.airway.order \leq e.order < |p.G'.E'|$  do
7     if  $e \in G_{HR}.E$  then
8       if  $u = null$  then
9          $u \leftarrow e.w_1$ 
10      else
11        if  $u = null$  then
12           $p_{aux}.G'.E' \leftarrow p_{aux}.G'.E' \cup e$ 
13        else
14           $v \leftarrow e.w_1$ 
15           $dist, E_{new} \leftarrow \text{Dijkstra}(G_{avail}, u, v)$ 
16          if  $E_{new} \neq \emptyset$  then
17             $p_{aux}.G'.E' \leftarrow p_{aux}.G'.E' \cup E_{new}$ 
18             $p_{aux}.G'.E' \leftarrow p_{aux}.G'.E' \cup e$ 
19             $u, v \leftarrow null$ 
20          else
21             $needsVertiport \leftarrow true$ 
22            break loop
23   if  $u \neq null$  then
24      $needsVertiport \leftarrow true$ 
25   if  $needsVertiport$  then
26      $source \leftarrow$  last node of  $p_{aux}.G'.E'$ 
27      $closest \leftarrow null$ 
28      $path_{best} \leftarrow null$ 
29      $dist_{short} \leftarrow \infty$ 
30     foreach  $vp \in \mathcal{V}$  do
31       if  $vp$  is not full then
32          $l_{vp} \leftarrow vp$  location
33          $dist, path \leftarrow \text{Dijkstra}(G_{avail}, source, l_{vp})$ 
34         if  $dist < dist_{short}$  then
35            $closest \leftarrow vp$ 
36            $dist_{short} \leftarrow dist$ 
37     if  $closest \neq null$  then
38        $p_{aux}.G'.E' \leftarrow p_{aux}.G'.E' \cup path_{best}$ 
39     else
40       send a message to  $p.d$  informing that there is no available path to fly
41   if  $p_{aux}.G'.E' - p.G'.E' \neq \emptyset$  then
42      $p \leftarrow p_{aux}$ 
43     ZSP-Drone-PP-Updt( $p$ )

```

If one of the two above conditions occurs, the algorithm chooses the best *vertiport* for the drone (Lines 25–40). Summarily, we calculate the shortest path from the drone

to each *vertiport* (since it has a “slot” to shelter the drone), selecting the one with the shortest distance (Lines 26–38). If there is no available *vertiport*, the cloud system emits a message to the drone informing that there is no available path to fly (Line 40). In this case, the drone service provider must take an external action unrelated to our mechanism. Finally, if the path was affected in one or more segments, the current course is replaced by the new one and submitted to the nearest ZSP to *p.d*, informing the drone about the update (Lines 41–43).

► *Time Complexity Analysis*: this algorithm has a series of simple assignments and variable initializations (Lines 3, 4, 5, 9, 12, 14, 17–19, 21, 24, 26–29, 32, 35–36, 38, 42), we state all of them as $\mathcal{O}(1)$. The main part of Algorithm 16 iterates over the defined $|d|$ path plannings (Lines 2–40). This iteration has an inner loop going through all the edges of the path planning in the worst case (Lines 6–22). Also, the size of edges in the path planning is limited to $|G.E|$, although it hardly happens in practical terms. Inside this inner loop, the Dijkstra call of Line 15 dominates the entire time complexity of the loop in the worst case. Hence, the inner loop has a time complexity of $\mathcal{O}(|G.E|(|G.V| + |G.E| \log |G.V|))$. As the airways topology of IoD is always a connected graph [5], $|G.E| \geq |G.V|$. Therefore, this time complexity can be reduced to $\mathcal{O}(|G.E|^2 \log |G.V|)$.

Besides this inner loop, the *vertiport* processing (Lines 25–40) has significant time complexity. This task iterates over all the *vertiports*, which is limited by $|G.V|$. Nonetheless, it hardly happens in real scenarios since it would mean that all the available points would be *vertipoints*. In this iteration, another Dijkstra algorithm dominates the time complexity. Hence, the iteration has a time complexity of $\mathcal{O}(|G.V|(|G.V| + |G.E| \log |G.V|)) = \mathcal{O}(|G.V|(|G.E| \log |G.V|))$.

Given this discussion, Equation 5.4 presents the time complexity of Algorithm 16:

$$\begin{aligned} T(\text{Cloud-Rfrmlt-PP}) &= |d|(|G.E|^2 \log |G.V| \\ &\quad + |G.V|(|G.E| \log |G.V|)) \\ &= |d|(|G.E|^2 \log |G.V|) \end{aligned} \tag{5.4}$$

Although Algorithm 16 presents a theoretical quadratic time complexity, it does not occur in practice since the path planning has several edges significantly less than the whole topological graph.

5.5 Simulation Setup and Performance Evaluation

We conduct an experimental evaluation through simulations to investigate the following aspects: (i) how JA affects drone mobility in the IoD environment; and (ii) how IoD-JAPM contributes to overcoming the challenges considering existing anti-jamming mechanisms. In this section, we present the evaluated scenario, the simulation parameters, and the metrics.

5.5.1 Compared Approaches

We consider four approaches to evaluate the proposed mechanism. We divide them into a scenario without JA (representing an optimum regarding the drone trajectory) and three scenarios where different solutions mitigate the JA.

- ▶ **Free-JA:** In this approach, there is no JA underway. Drones will communicate and move as much as possible following the airways. This approach represents an “optimum” scenario in terms of flight time and power consumption.
- ▶ **Baseline mechanism** [157]: This approach focuses on our previous mechanism to detect and mitigate a long-term JA, performed by a malicious entity on the ground, keeping stationary mobility.
- ▶ **Joint Optimization Approach** [51]: An important aspect when analyzing the robustness of IoDJAPM is to compare our solution with a state-of-the-art method. However, these methods consider the airspace as free-to-fly, differing from our assumptions. Thus, we need to consider an approach as similar as possible to evaluate a fair comparison. Bearing this in mind, we consider the approach proposed by Wu et al. [51], denoted as “Joint”. The proposed method aims to optimize jointly the communication throughput and the UAV’s mobility. The UAVs also move towards well-defined locations: the targeted grounded nodes. Considering our proposed scenario, they can be taken as terrestrial waypoints of the UAV’s trajectory.

Nonetheless, the main difference between the approaches is airspace navigability. Hence, we adapt the proposal of Wu et al. [51] so that the drone trajectory remains inside the airway’s boundary radius and, in a JA situation, the “unused airspace” (as described in Figure 5.4).

We define new constraints regarding the mobility capability of the UAV to meet this adaptation, presented in Equations 5.5, 5.6, and 5.7.

$$\text{Min}_{dist}(\mathbf{q}_u[i], \beta) \leq r \quad (5.5)$$

$$\mathbf{q}_u[i]^z \leq \alpha \quad (5.6)$$

$$\text{distance}(\mathbf{q}_u[i], w) \leq \text{distance}(\mathbf{q}_u[i-1], w) \quad (5.7)$$

In Equation 5.5 we define that the minimum distance between the position \mathbf{q}_u of a given drone u at the moment i and a given line segment of an airway β must be less or equal to a radius r . In other words, the calculated position must be inside the boundaries of the airway β . In Equation 5.6 we define that the calculated altitude (corresponding to the z-axis on the Euclidean system) must be less or equal to the maximum available altitude α .

Since the proposal [51] aims to jointly optimize the drone trajectory and the throughput with a targeted node, the calculated positions tend to approximate drones to the ZSP instead of the defined waypoint. To avoid this case, we define in Equation 5.7 that the distance between the calculated position of the drone and the waypoint w must be less than the previous distance. Thus, we ensure that the drone moves toward the waypoint instead of the ZSP. All these constraints are valid and applicable to all the assumptions, equations, and algorithms presented by Wu et al. [51].

► **IoDJAPM**: This approach is similar to **Baseline**, but the network operates with IoD-JAPM to mitigate the JA impact. Hence, it is possible to compare the performance of **IoDJAPM** with both the former approaches, namely **Baseline** and **Joint** [51] methods.

5.5.2 Simulation Parameters

We consider two distinct airway topologies, \mathbf{T}_1 and \mathbf{T}_2 , summarized in Table 5.1. \mathbf{T}_1 is a robust infrastructure where drones have parallel airways and different flying paths, designed following a piece of the roadside's structure in Manhattan Island, NY. In turn, \mathbf{T}_2 is a constrained infrastructure, where drones have limited paths to fly at a single available altitude. This topology follows a piece of the roadside structure of San Francisco, CA. Also, \mathbf{T}_1 has two available *vertiports* with a double capacity as the single *vertiport* of \mathbf{T}_2 .

Table 5.1: Topology Attributes

Attr.	T₁	T₂
Region Size	$3 \times 3 \times 0.3 \text{ km}^3$	$4 \times 2 \times 0.1 \text{ km}^3$
#available altitudes	3	1
#Airway nodes (total)	402	60
#Airway segments (total)	997	98
# <i>Vertiports</i>	2	1
<i>Vertiport</i> capacity	4	2

We performed the simulations using the IoDSim. Table 5.2 shows a list of simulation parameters, mainly regarding the environmental nodes. These parameters are defined considering a realistic scenario with a reasonable time for an attacker to perform JA [51]. Likewise, the distributed system can detect and act against the attack.

We evaluate the scenarios with one and two jammers. In both cases, the first jammer has a low interference signal range, affecting only the airway at the lower altitude. The second jammer, in turn, has a wide interference range, performing a JA over all the parallel airways of an affected region.

Each simulation has 20 minutes. We set 25 drones following a Gauss-Markov mobility pattern with speeds varying from 5 to 10 m/s. The Gauss-Markov model is configured with an angle $\alpha = 0.75$ over the navigation vectors. Thus, a given drone flying toward an aerial waypoint has small deviations in its flight, simulating potential weather conditions in the environment, such as the wind performance. We spread four ZSPs on the ground communicating with both the drones and the cloud system. Regarding communication, the protocols and radio configuration are the same for all nodes. The designed framework considers UDP, AODV, and CSMA/CA as the transport, routing, and MAC protocols, respectively. The radio has an APSK modulation, a frequency of 2.4 GHz, and a transmission power of 220 mW. The SNIR threshold is set as 4 dB.

For each combination of scenario \times topology \times number of jammers (except for Free-JA with Jammers), we conducted 35 experiments with distinct seeds and path planning of drones, leading to results with a confidence interval (CI) of 95%. The Jammer and ZSPs positions were the same for all replications. Hence, each experiment reflects a different drone mobility scenario over the same attackers' positions.

5.5.3 Metrics

We consider four metrics to measure how JA affects the drone path planning, focusing on the topological airways affected by the JA, the number and type of affected

Table 5.2: Simulation Parameters

Parameter	Value
Scenarios	Free, Baseline, Joint, IoDJAPM
#Jammers	{1,2}
Topologies	{T ₁ , T ₂ }
Simulation time	20 minutes
Drone speed	Uniform [5–10] m/s
Mobility pattern	Gauss-Markov ($\alpha = 0.7$)
#Drones	25
Jammer's mobility	Grounded, Stationary
Jammer's behavior	Long-term attack
#ZSPs	4
ZSP update check interval	1 s
ZSP Update check threshold	5
Transport Protocol	UDP
Routing Protocol	AODV
MAC Protocol	CSMA/CA
Radio's Modulation	APSK
Radio's Frequency	2.4 GHz
Radio's Bandwidth	2 MHz
Radio's Transmission Power	220 mW
SNIR threshold	4 dB
Convergence threshold μ (Joint[51])	10^{-3}

planning, the increasing flight distance, and the increasing power consumption. They are described as follows.

► **Hazard Region Rate (HRR):** *HRR* measures how large the HR is compared to the topological graph. Equation 5.8 formally describes how to calculate *HRR*, where G_{HR} is the graph corresponding to the HR and G is the topological airway graph of the network.

$$HRR = \frac{|G_{HR}.E|}{|G.E|} \quad (5.8)$$

► **Drones with Affected Path Planning Rate (DAPPR):** This metric indicates the rate of drones whose path planning was affected at least once due to JA compared to the total number of drones. We consider affected path plannings those routes that need to be reformulated, redirected to a *vertiport*, or those that could not proceed to a final destination. Hence, it does not embrace the flights that could proceed after a successful airway analysis. Formally, Equation 5.9 defines *DAPPR*. $|\mathcal{D}_{affected}|$ represents the total number of drones with an affected path planning, such that the ones reformulated, redirected to a *vertiport*, or that could not proceed to the final destination.

$$DAPPR = \frac{|\mathcal{D}_{affected}|}{|\mathcal{D}|} \quad (5.9)$$

► **Increasing Flight Distance Rate (IFDR):** *IFDR* calculates the relationship between the total distance of the Free-JA scenario and the corresponding path planning in the scenarios with an anti-jamming mechanism. Equation 5.10 defines *IFDR*, where PP_o is the original path, PP_c is the corresponding path in the Baseline or IoD-JAPM scenario, and $dist()$ is a function that calculates the total distance of a given path.

$$IFDR = \frac{dist(PP_c)}{dist(PP_o)} \quad (5.10)$$

► **Increasing Power Consumption Rate (IPCR):** Similar to *IFDR*, this metric compares the relation between the original and the corresponding reformulated path planning in terms of power consumption. Equation 5.11 describes *IPCR*, where \mathcal{E}_o is the power consumption of the drone considering the Free-JA scenario, and \mathcal{E}_r is the power consumption of the corresponding drone with the reformulated path planning.

$$IPCR = \frac{\mathcal{E}_r}{\mathcal{E}_o} \quad (5.11)$$

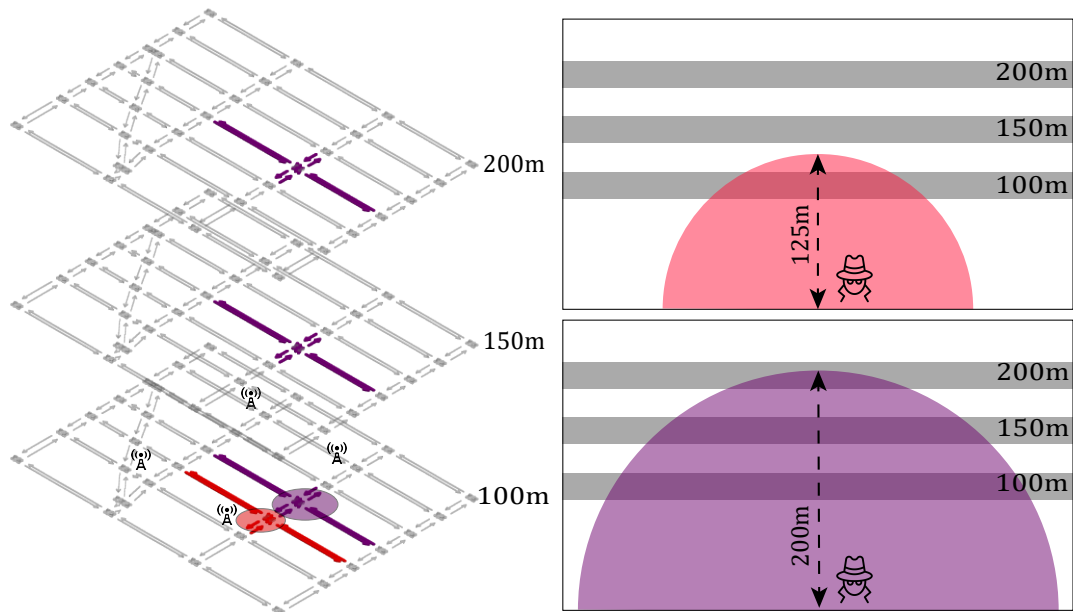
5.6 Results and Analysis

This section presents the results obtained through the simulations. We discuss each metric's results thoroughly, describing how IoD-JAPM mitigates the impact of JA, promoting a safer IoD environment. For all results, the interval errors are less than 1 measure unit. Therefore, they are not represented in the charts.

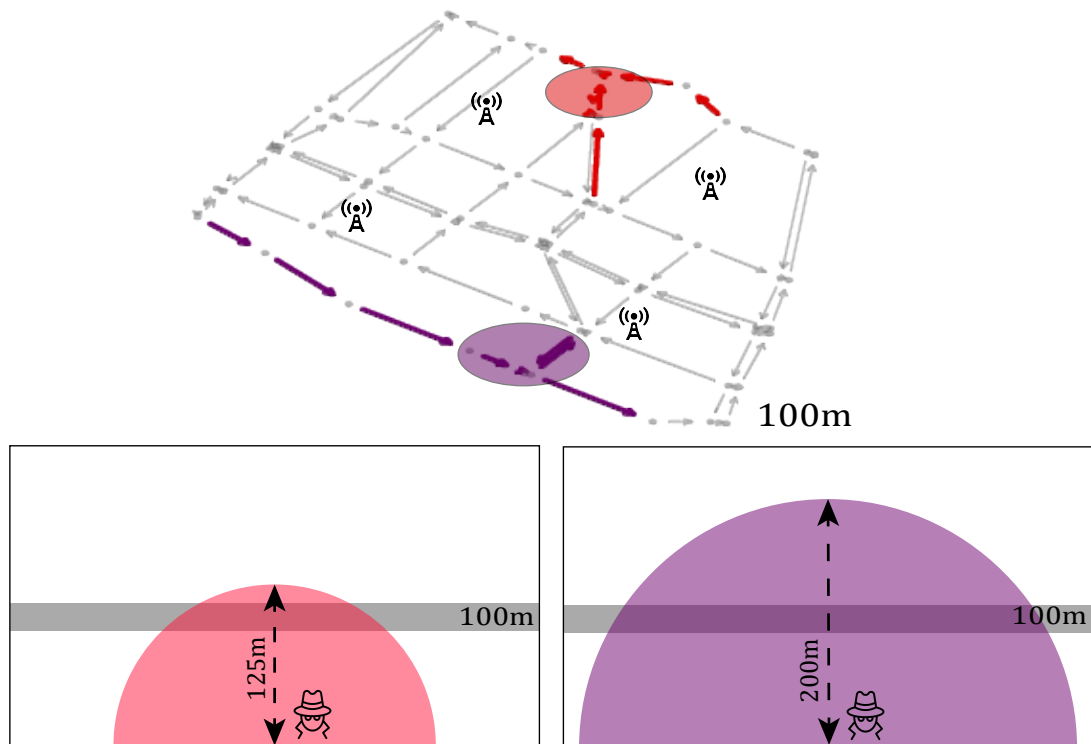
► **HRR Analysis:** Understanding the dimension and how HR affects airspace is the first door to analyzing the impact of JA over the IoD. Figure 5.6 shows the calculated HR for T_1 (Figure 5.6a) and T_2 (Figure 5.6b), considering the region of the two jammers. Regarding Figure 5.6a, the overlapped airways are out of scale compared to the other elements to facilitate the visualization. Also, the grounded elements (ZSPs and the region over the JA) are represented together with the lower airway.

Elements with red color refer to the first jammer (the one with less “jamming power”), and the purple color refers to the second jammer. Hence, the red circle indicates the region over the active JA from the first jammer, while the purple circle refers to the JA from the second jammer. Considering that these circles illustrate only a slice of the jammed region on the illustration plan, we provide a vertical visualization of these regions, which explains why the second jammer can affect all three altitudes of T_1 . Likewise, the red and purple airway segments correspond to the HR calculated following Algorithm 15.

Figure 5.6: The typical Hazard Region calculated in each topology over all the airways



(a) Hazard Region of the topology T_1

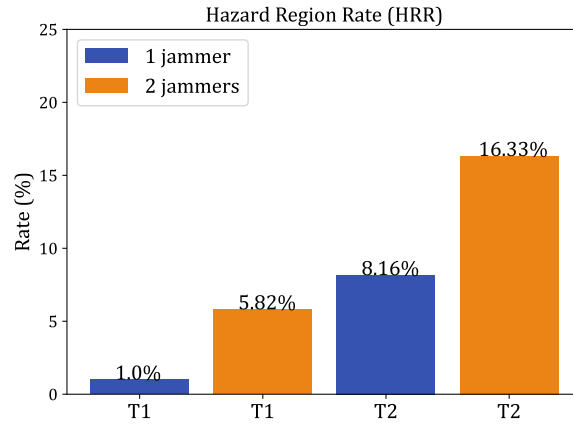


(b) Hazard Region of the topology T_2

Source: Elaborated by the author

Figure 5.7 presents the results of HRR , giving a numerical meaning to the topological representations. The lower the HRR , the lower the impact on the IoD environment. The robustness of the topology is a determining factor in the HR coverage over the environment. Even reaching the three parallel altitudes, the HR caused by the two jammers

Figure 5.7: Results of Hazard Region Rate regarding IoD-JAPM performance evaluation



Source: Elaborated by the author

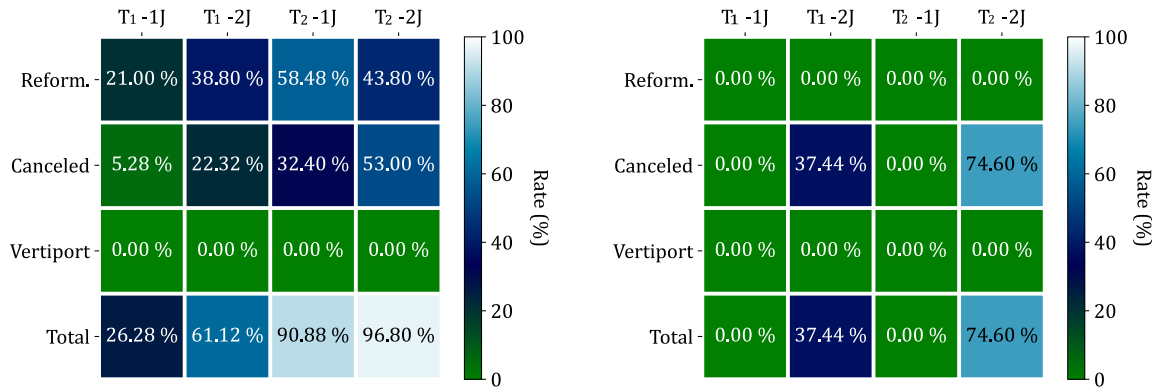
acting together over T_1 is less than the HR over T_2 caused by a single jammer. Indeed, the parallel airways of T_1 provide a drone with several ways to avoid JA, as reflected in the *HRR*. Furthermore, even though the *HRR* of T_2 does not appear to be such a high rate initially, it significantly restricts the airspace for drones, impacting their path planning directly. In the following sections, we discuss these aspects in detail.

► **DAPPR Analysis:** The *DAPPR* metric gives a profile of the trajectories in each evaluated configuration. Figure 5.8 presents these results for the evaluated approaches: baseline in Figure 5.8a, the Joint approach [51] in Figure 5.8b, and IoD-JAPM in Figure 5.8c. Each cell represents the *DAPPR* of a given type of path planning (lines) affected in a given configuration (columns). The last line presents the sum of the rates of the configuration rates: the less the total *DAPPR*, the less the impact of JA over the IoD environment.

The total rate of affected path planning highlights that IoD-JAPM can mitigate the impact of JA at a higher level than the baseline and Joint [51] approaches. For the scenarios with one jammer, IoD-JAPM prevented all drones from being attacked during all experiments. Compared with the baseline mechanism, the rate is higher than 90% on average for the topology T_2 . These aspects reveal how vital the altitude analysis step is, even considering a restricted topology.

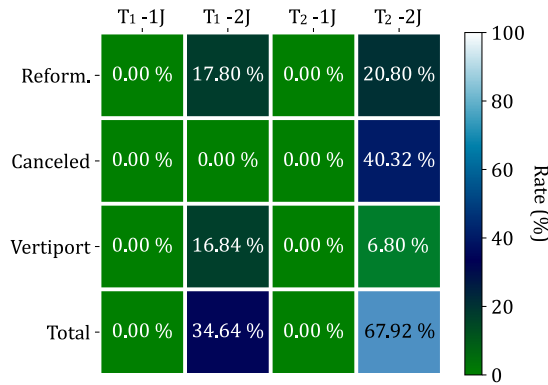
Observing the results related to the presence of two jammers, IoD-JAPM overcame the Joint approach [51], especially in Topology T_2 . Nonetheless, Figure 5.8b shows the limitations of the Joint approach [51] in a restricted scenario with well-defined airways. In this case, the flights were canceled because the drone position must remain in a range based on the drone maximum speed. Therefore, the mechanism can not find a possible trajectory without violating the imposed constraints [51]. Indeed, for the topology T_2 , with two jammers, almost 75% of the drones suffered from this issue, on average. As

Figure 5.8: Results of Drones with Affected Path Planning Rate regarding IoD-JAPM performance evaluation



(a) Baseline mechanism

(b) Joint approach [51]



(c) IoD-JAPM

Source: Elaborated by the author

presented in Figure 5.6b, the jamming range of the second jammer in T_2 is stronger than the first one, which can explain the obtained results.

Also, the use of *vertiports* mitigates the impact of JA, avoiding the cancellation of a significant portion of path planning compared to the baseline. For the topology T_1 , it was not necessary to cancel any path planning, while for T_2 there was a decrease of 25% from the corresponding scenario with the baseline mechanism. This aspect is strictly related to the robustness of the topology. While in T_1 , there are two available *vertiports* (which can assist four drones each), T_2 has only one *vertiport* (assisting two drones only). Indeed, in most experiments regarding T_2 with two jammers, the *vertiport* was full in most experiments.

Therefore, these results reveal a fundamental challenge: even with IoD-JAPM showing adequate protection against JA, the topology of the airways is still a risk factor to be considered in IoD. When such a topology has few available paths for the drone flight and does not offer a reasonable number of available flight altitudes, attackers with greater

Figure 5.9: Results of Increasing Flight Distance Rate regarding IoD-JAPM performance evaluation

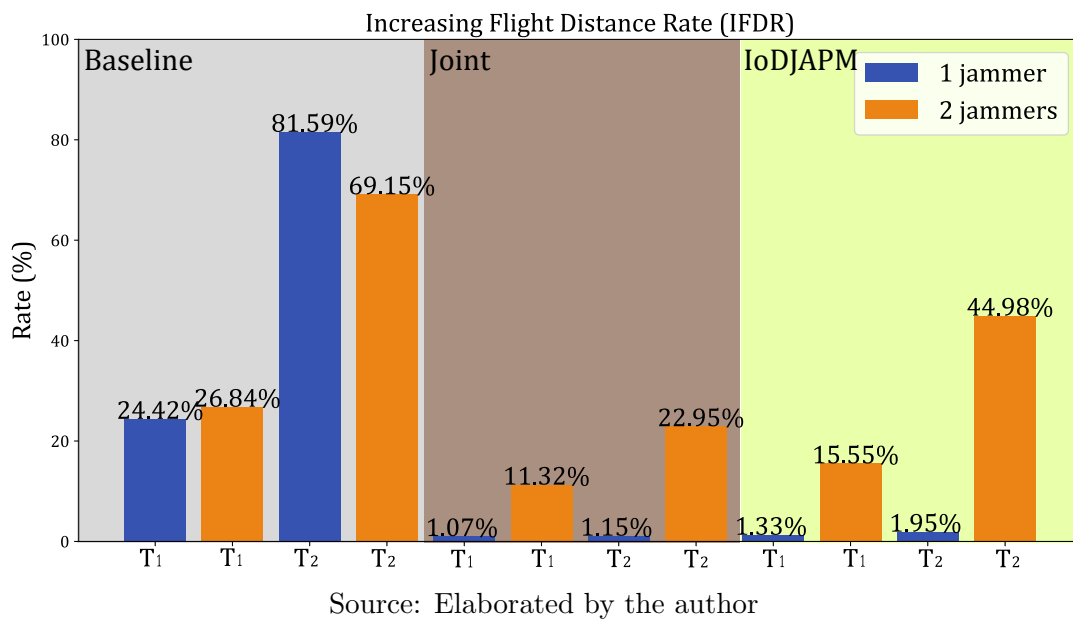
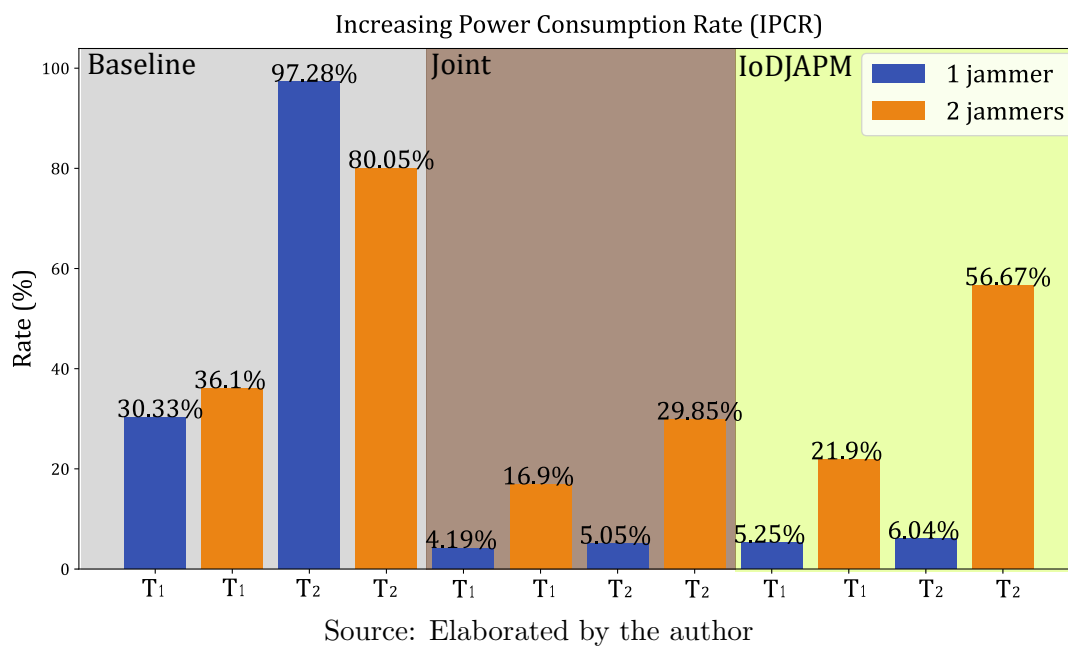


Figure 5.10: Results of Increasing Power Consumption Rate regarding IoD-JAPM performance evaluation



interference power can significantly impact the drone trajectory and, consequently, the provided service.

► **IFDR Analysis:** As we discussed previously, drones have SWaP limitations, being essential to assess factors that can lead to the excessive expenditure of these resources. One of these factors is the increasing flight distance caused by applying the anti-jamming mechanisms measured by the *IFDR*. The less the *IFDR*, the less the impact on the IoD.

Figure 5.9 summarizes the obtained results regarding this metric.

IoD-JAPM overcomes the baseline mechanism in all scenarios. Indeed, IoD-JAPM has an increase significantly less than the baseline and a similar rate compared to the Joint approach in scenarios with one jammer. In these cases, the slight increase occurs due to the altitude analysis step of IoD-JAPM, in which the drone performs minor deviations from its original path, as presented in Figure 5.5. As there is no other reformulation, this is the only kind of increase.

Analyzing the results of Topology T_2 with two jammers, IoD-JAPM increased twice the flight distance compared to the Joint approach [51]. This behavior occurs mainly because there is no reformulation of path planning on the Joint approach. Therefore, the trajectory deviations are restricted to the maneuvers performed by the drones to optimize the throughput with the ZSP communication. Indeed, if we analyze this result allied with the DAPPR (Figure 5.8), IoD-JAPM still is the best mechanism against JA since it cancels fewer flights than the Joint approach, reformulating a considerable amount of them as well as sending drones to vertiports.

Still considering the results of *DAPPR*, the altitude analysis performed by IoD-JAPM explains the better results obtained. The rate of reformulated path planning is always less than the baseline mechanism. Even so, some path plannings reformulated by the baseline mechanism did not change when IoD-JAPM was applied but proceeded normally because the altitude analysis was successful.

Furthermore, we can note an outlier in the results of T_2 of the baseline mechanism: the *IFDR* of one jammer was higher than two jammers, going in the opposite direction than expected. Analyzing the results carefully, we note that it occurs due to the decrease of reformulated path plannings from one jammer to the scenario with two jammers. A significant portion of flights was canceled when two jammers were in the environment, as presented in Figure 5.8a. Some of these flights were reformulated in the presence of a single jammer. This reformulation allows the drone to cross the entire environment since the second jammer region is on the opposite side. Therefore, these trajectories contribute significantly to the increasing distance, but they do not appear in the scenario with two jammers since they are canceled.

► ***IPCR* Analysis:** Figure 5.10 presents the results of *IPCR*. The less the *IPCR*, the less the impact on the IoD, specifically, on the drone battery efficiency. In summary, *IPCR* is interlaced with the results of *IFDR*. We can note a similar pattern in the bar results, comparing the charts in Figures 5.9 and 5.10. These patterns can be identified because the movement of the drone propellers affects energy consumption more than the transmission and reception of wireless messages [135]. Hence, the higher the *IFDR*, the higher the *IPCR*.

Likewise discussed in the previous results, IoD-JAPM overcomes the baseline mech-

anism in all scenarios, with discrete increasing power consumption for scenarios with one jammer. Nonetheless, the *IPCR* is higher than 50% in T_2 with two jammers. Although the rate of reformulated path planning is not higher (about 20% on average), the flights redirected to *vertiports* sustain a significant power consumption.

Comparing IoD-JAPM with the Joint approach [51], the results follow the same behavior presented in *IFDR*, which can be explained by the same reasons. Although the power consumption of IoD-JAPM is considerably higher in the more restricted scenario, the *DAPPR* must be taken into account, where IoD-JAPM overcame the Joint Approach in all scenarios.

These aspects reinforce the challenge imposed by restricted topologies. Even with IoD-JAPM, the lack of available pathways directly impacts on the distance that drones fly to complete their itinerary and, therefore, the power consumption. To mitigate these factors, *vertiports* can be placed in strategic regions to cover the airspace optimally. However, this placement involves several aspects since they are commonly deployed in buildings or dedicated areas. All in all, new studies must be conducted to investigate it properly.

► **General Discussion:** The drone traffic control through well-defined airways is undoubtedly fundamental to IoD. With the constantly growing of drone-based services, it is mandatory that a network authority manages and controls the drone flight, imposing flyable boundaries. Although the concept of airways promotes several advantages, it represents a risk to avoiding JA, severely affecting drone performance.

In a nutshell, the obtained results pointed out a deep relation between the impact of JA on the drone path planning and the robustness of the airway topology. The more restricted the topology, the more impact JA causes in the IoD: the drone trajectory deviates considerably compared to its original path, increasing the flight distance and power consumption.

IoD-JAPM can mitigate this impact, overcoming our baseline mechanism in all scenarios considering the evaluated metrics. Indeed, IoD-JAPM significantly decreases the number of affected path plannings (mainly those canceled due to the region affected by JA), the increasing flight distance, and the power consumption of the reformulated path plannings.

Regarding the approaches designed primarily for airspace “free-to-fly”, the performed drone maneuvers can outperform IoD-JAPM regarding the flight distance and power consumption in scenarios with restricted topology. Nonetheless, these approaches can not provide a valid trajectory when the jamming signal is strong, as pointed out by our comparative evaluation with the Joint approach [51].

Restricted airway topology against robust jammers represents a profound challenge that must be investigated further. Even with the application of proper anti-jamming mechanisms, an IoD airway topology with few available aerial pathways and few parallel

altitudes is affected by strong jammers severely. Depending on the strategic position of these attackers, the protection mechanism (e.g., IoD-JAPM) can not mitigate the attack correctly, and, therefore, the network can be compromised, affecting the drone flight.

As cars can wait on the roadsides when an accident occurs, drones may have a proper place to stay until the network recovers from the attack. *Vertiports* are proper PoIs to serve as waiting regions. However, they must be placed to optimize access and drone allocation, involving several factors. Hence, the *vertiports* coverage and placement are open challenges to investigate further.

5.7 Chapter Remarks

In this chapter, we presented our contributions regarding anti-jamming mechanisms for IoD. We investigated the impact of JA on drone path planning, and, therefore, the drone trajectory on the IoD. We discussed thoroughly that existing solutions could not adequately meet one of the main IoD characteristics: drone traffic control through well-defined airways, where the drones fly over constrained airspace.

Hence, we proposed the IoD-JAPM, an airway-aware protection mechanism against JA on the IoD. This mechanism can mitigate the JA impact by searching for an available altitude to communicate with the ZSP without JA interference. To address it, our solution takes advantage of the unused airspace between two or more parallel altitudes over different airways without violating the airways boundaries, avoiding collisions with other drones or obstacles (e.g., buildings). Also, IoD-JAPM redirects the drones to *vertiports* when generating a new path planning is not possible due to JA, keeping the drone safe until the attack is adequately neutralized.

We conducted a performance evaluation through simulations, comparing IoD-JAPM with a baseline solution and an adapted existent approach [51] that considers the airspace free to fly. We considered environments with different airway topologies and a different number of jammers performing attacks. IoD-JAPM overcomes the baseline solution in all scenarios, mitigating the effects of JA over the path plannings, causing few reformulations or cancellations. Furthermore, IoD-JAPM causes a slight increase in the drone's original flight distance, leading to better power consumption management. Compared to the "free-to-fly" approach [51], IoD-JAPM presented a similar increase in flight distance and power consumption for robust topology. In constrained scenarios, IoD-JAPM is outperformed in these characteristics, which is an expected behavior since the "free-to-fly" approach does not reformulate path planning. This issue directly impacts the drone trajectory in environments with high jamming signals, where the method can

not process a valid trajectory deviation, leading to several flight cancellations. Thus, our evaluation reinforces that these approaches can not fit properly with the deployment of IoD environments following well-defined airways.

Therefore, we are able to answer the research questions of this dissertation considering the design of anti-jamming mechanisms for IoD:

► **Can the existing anti-jamming mechanisms provide the same protection level to IoD environments when compared to traditional mobile networks?**

Answer: Taking into account IoD environments with the presence of well-defined airways, they can not. As discussed in this chapter, airways pose boundaries to the full application of the existing strategies, in such a way that the provided protection can not ensure a suitable security level for drones in terms of the completeness of their path planning.

► **Is it possible to adapt these existing anti-jamming mechanisms to enhance the protection level provided to a given IoD environment?**

Answer: Yes, it is possible. We demonstrated that well-known JA detection strategies must be integrated with novel reformulating path planning techniques, as occurs in IoD-JAPM. The extensive performance evaluation highlighted that the proposed mechanism can provide a suitable level of protection against JA, mitigating its effects on drone path planning.

Also, the results reveal novel challenges in this field. IoD environments with restricted airway topology (i.e., few available pathways and few parallel altitudes to fly) can be severely impacted by JA, even with the application of IoD-JAPM. This impact is related to the strategic position of the attacker. An interesting way to mitigate this impact is to model strategies for the optimal deployment of *vertiports*. In this strategy, the *vertiports* are deployed in points to optimize the environment coverage and the drone flight distance.

In future directions, we plan to explore the opportunities and investigate the current challenges, summarized as follows.

- Design novel strategies to optimize the deployment of *vertiports* and energy-related constraints, for instance, based on ML approaches;
- Study the computational, financial, and legal factors to deploy IoD environments with robust airway topology;
- Apply the mechanism in a variety of topologies to evaluate the provided protection in a broader scope;
- Apply and evaluate IoD-JAPM in real drone-based mobile environments.

Chapter 6

Design of Automatic Drone Detection Strategies for IoD

This chapter presents the contributions regarding the design of ADD strategies, specifically, the study about the use of a drone's propeller acoustic signal as an input source, and the introduction of the dissimilarity concept to detect unknown UAV signals, resulting in a smart strategy to detect drone's through rhythmic-based features and a new detection mechanism, named DissIdent. These contributions correspond to initial findings regarding the use of rhythm properties, and the detection through dissimilarity techniques, respectively. Hence, they make room for future research in this direction.

Section 6.1 brings an introduction and the motivations regarding the design of novel ADD strategies, highlighting the current challenges. We present the application scenario and threat model in Section 6.2. The related studies are discussed in Section 6.3. Our first related contribution, the design of a smart rhythm-based strategy, is presented in Section 6.4 in which we formally model the strategy, carry out a performance evaluation, and discuss the related results. Likewise, we present our second related contribution, the DissIdent approach, in Section 6.5, including its formal definition, the performance evaluation, and the related results. Lastly, section 6.6 presents the chapter remarks, in which we summarize our main findings and discuss some directions to improve the investigated research fronts.

6.1 Introduction

As discussed in Section 3.2.4, Automatic Drone Detection (ADD) is the task of correctly detecting the presence of a drone given a set of environmental features. Moreover, identification is a task that can be stemming from the detection. Identification can lead to detailed information about a detected drone, such as its model, size, weight, and speed [92]. ADD can protect IoD nodes from unauthorized/unknown entities, being a

silent and supportive mechanism for other PMs (e.g., anti-jamming). Therefore, the investigation of both tasks represents fundamental aspects to ensure security and privacy in the IoD environment.

Over the last few years, ADD has been improved with the exploration of smart approaches integrated with traditional technologies and methods, such as the use of Deep Learning algorithms [92, 93, 161]. Acoustic-based features have been widely applied in these strategies, but they are mostly based on well-known aspects, such as Doppler and frequency-related signatures [92]. However, there is room for the investigation of other acoustic-related characteristics. Rhythm-based descriptors are powerful features extracted from audio-based signals that express rhythmic properties, being useful in different tasks, e.g., music and movie genre classification [162]. The drone's propellers produce a prominent sound in the environment. Also, different drone models have different propeller engines, presenting specific working behavior [1]. Therefore, this signal can be explored through rhythmic properties.

Although the current AI-based strategies address a suitable accuracy, the detection is dependable on known data. In other words, the approaches can properly detect authorized drones since they have a pool of samples from the expected categorization, but they fail to detect unknown entities and even distinguish different unauthorized UAVs. From the IoD network point of view, grouping unknown UAVs can enhance the knowledge about the environment since they can represent malicious drones, being valuable information for defining strategies to avoid them.

Clustering models can handle this challenge. They group the observed data based on their similarities given a measuring space and do not require a pool of data labeled *a priori*, grouping them dynamically [163]. However, large-scale datasets represent a challenge to the application of traditional clustering methods due to their computational complexity as well the potential high dimensionality of the data [163].

Also, the uncertainty regarding different unauthorized drones can lead to a large multi-class problem, being unfeasible by these traditional methods. Recently, dissimilarity techniques presented relevant results for independent-class problems. The dissimilarity concept is a classification-centered approach based on the (dis)similarities between patterns to distinguish one from the other, grouping unknown elements [164]. This approach can identify patterns from different features, such as acoustic and RF-based signals. Hence, this approach is a proper alternative to handle these challenges.

In this chapter, we introduce to new front of studies regarding ADD applied to the IoD environment:

- The first front is the investigation of rhythmic-based acoustic features for ADD tasks, extracted from the drone's propeller sound. To analyze and validate our investigation, we conduct a case study considering different rhythm-based descriptors, using the public and freely available dataset of drone and non-drone acoustic signals;

- The second front investigates dissimilarity techniques for the detection of unknown signals in IoD environment. Therefore, we propose DissIdent, a dissimilarity-based approach for detecting unknown UAVs in the airspace. This solution can identify patterns from different features through a smart workflow involving ML and clustering concepts. We design an architecture that can distinguish patterns from input signals, grouping unknown elements regardless of the data dimensionality. Also, we formally define how to represent a given UAV-based feature in the dissimilarity space. Through extensive experiments, we demonstrate that DissIdent can overcome both supervised-based and clustering models regarding the detection and identification of known and unknown signals in the airspace, mitigating the trade-off between the tractability and accuracy of multi-class problems.

6.2 Application Scenario and Threat Model

Summarily, any IoD-related environment is a proper recipient to the deployment of ADD strategies, specifically, the DissIdent approach. For instance, drones will provide services to the population to improve people’s lives (e.g., drone delivery). Thus, it may be common to see a flow of drones in the city’s sky. Therefore, it will be necessary to use mechanisms to detect rogue and unauthorized drones. Different areas, such as airports, government buildings, and private companies may require the identification of unauthorized drones. Likewise, the industry shall require ADD mechanisms to keep its environment safe and free from malicious aerial entities.

The threats faced by ADD are more general than the ones presented in the previous chapters since ADD can primarily counter a wide range of attacks, as presented in Table 3.4. Nonetheless, we assume that a potential attack is underway when a set of unknown drones are flying in regulated airspace. We describe the related threat model as follows.

- The set of adversaries \mathcal{A} is divided into two groups: a set of ground malicious entities $\mathcal{A}_{ground} \subset \mathcal{A}$; and a set of aerial entities $\mathcal{A}_{air} \subset \mathcal{A}$. \mathcal{A}_{ground} embraces the human/cyber tools that control the aerial entities \mathcal{A}_{air} . This second group, in turn, is composed of drones flying over the airspace;
- The adversaries \mathcal{A}_{air} can perform two types of attack: passive attacks, such as EA; or active attacks, such as SA. Therefore, this model relates to mobile threats over the air;

- The devices related to *Aair* fly through an opportunistic behavior, focusing on performing the intended attack over a set of targets, generally authorized drones in the network. Once it is completed, they land. Thus, the window size to detect these threats can be small;

Although the IoD network can potentially detect and counter these threats by other mechanisms (e.g., based on authorization protocols or anti-jamming techniques), our focus delves into the primary detection of unknown entities in the airspace despite of what are their intentions, or even what kind of behavior they present. In other words, we consider the proposed ADD strategy as a “first barrier” to warn the whole network about a potential threat.

6.3 Related Studies

A seminal study of automated-based drone identification was presented by Moses et al. [165], in which they proposed a lightweight radar system for small UAVs, differentiating the drones through Doppler signatures. This system was attached to a monitoring drone, allowing an “in-flight” drone identification instead of a stationary, one. A well-explored phenomenon to detect drones is RF. Considering the advent of 5G communication, Solomitckii et al. [166] investigated the exploitation of 5G millimeter-wave signals, exploring the frequency and bandwidth. Nemer et al. [93] explored RF signals identifying and detecting UAVs through a hierarchical Machine Learning method.

Acoustic signals are also an explored phenomenon to detect drones smartly. Yang et al. [167] accomplished a solution for Direction of Arrival (DOA) problem considering the acoustic harmonic drones’ signals, integrating the model with a real-life drone tracking platform. Kolamunna et al. [92] proposed the DronePrint, a detection framework that uses drone acoustic signatures. Mandal et al. [168] modeled a low-cost system for detecting and classifying drones using five acoustic descriptors: Mel frequency cepstral coefficients, euclidean-based distance spectrogram, chromagram, spectral contrast, and tonal centroids. For classification, the SVM classifier was also used. There is also no information on the dataset’s availability in the study. Al-Emadi et al. [169] introduced a public and freely available dataset of acoustic signals of drones and non-drones. In the study, the audios are processed as visual representations of the spectrogram and used as input data for three different modeled Deep Neural Networks: a CNN, RNN, and a CRNN. The classifications were carried out on two approaches: a binary classification to detect if the acoustic signal was or was not a drone; and a multiclass classification to

consider the audio as not being from a drone, or being a “Bebop” or “Mambo” model drone.

Differing from the aforementioned studies, Unlu et al. [170] designed a drone detection system based on the drone’s captured images. Several shape descriptors were extracted from the pictures and modeled to a Neural Network to classify them. An image database of drones and birds was considered, in which the system was able to differ from these two categories. Recently, the study of Sciancalepore et al. [26] presented the PiNCH, a framework for drone detection that combines different network traffic information with machine learning classic algorithms, such as Random Forest. This study considers traffic analysis as its primary method to accomplish the detection. Svanstrom et al. [171] proposed a framework to detect drones based mainly on thermal signatures captured by thermal cameras, supported by GPS and radar signals.

As we can note, none of them explores the hypothesis of the drone’s acoustic signature having rhythmic properties. Hence, our proposal pursues to empower drone identification through rhythmic acoustic features, exploring and extending the field of possible characteristics to be extracted from the acoustic signals of drones. Furthermore, although these approaches considered realistic sources as non-drone/non-authorized signals, they did not follow up with this categorization in the testing phase. In other words, they fail in the identification task of unknown nodes. Nonetheless, the airspace is susceptible to a plethora of external signals, mainly in urban scenarios, where unknown drones can be detected.

Clustering models can tackle this issue. These strategies are well-applied in noisy and large datasets [163]. Some algorithms addressed good performance in terms of detection of outliers and independence of the order in which the strategy receives the data as input. The literature points out three strategies considering this context [163]: Density-Based Spatial Clustering of Applications with Noise (DBSCAN); Ordering Points To Identify the Clustering Structure (OPTICS); and the Balanced Iterative Reducing and Clustering using Hierarchies (BIRCH).

Signals coming from authorized drones certainly will be near each other considering a given representation space. Likewise, signals from unknown entities are potentially far from the latter group, representing outliers, or even closer to each other if they come from the same unknown source. Although they are not commonly applied in UAV detection, they represent a proper solution to identify both known and unknown signals. However, signals coming from different sources over the airspace can lead to a problem with high dimensionality, representing a challenge to clustering models [163].

DissIdent can overcome these challenges by embracing both classification and clustering strategies. Our approach takes advantage of the dissimilarity concept, distinguishing the input signals through a smart identification workflow using learning classification models and clustering techniques.

6.4 Design of a Smart Rhythm-Based Strategy for ADD

Each drone device model is singular with a unique hardware configuration, and, consequently, it generates individual acoustic signals based on its propellers and motors performance. Informally, we can assume that each drone has its own “rhythm of flight”. Bearing this in mind, we investigate the following hypothesis: *The acoustic signal generated by different drone flights has distinct rhythmic properties that can improve the drone identification task.*

In this section, we propose a new methodology based on rhythmic features for ADD, extracted from the drone’s flight acoustic signals. Figure 6.1 shows the proposed scheme. From the acoustic signal of a sample, we extract rhythm-based features. Aiming to combine these features with descriptors from a different nature, we also generate the audio’s spectrogram, extracting a visual-based descriptor. These features are then combined themselves and also with the visual descriptor. After, they are given as input to a DNN classifier. The methodology allows for two types of feature combinations: early fusion (before given as input to the classifier) or late fusion (combines the classifier results with different features), detailed further.

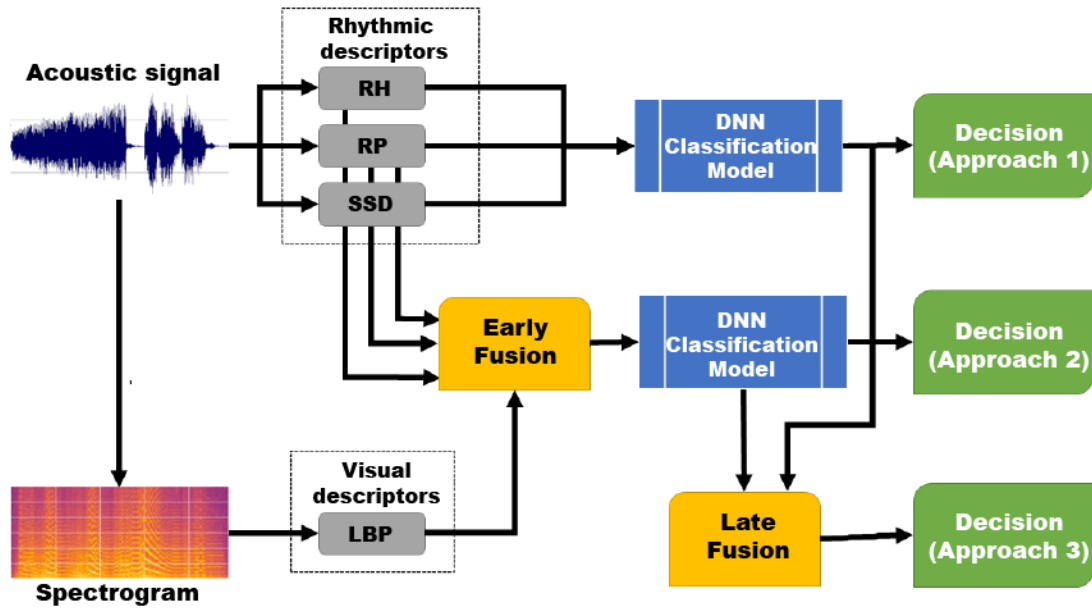
We evaluate three different decision approaches, described as follows.

- The decision is taken considering the classification results of a rhythmic descriptor, solely;
- The decision is taken considering the classification results of an early fusion configuration, which can be composed of both rhythmic and visual descriptors;
- The decision is taken combining different classification results – coming from the previous approaches – and submitting them to a late fusion rule, which will generate the final decision.

6.4.1 Feature Extraction

In different fields, rhythmic features are commonly combined with descriptors from a different “nature”, mainly visual features. For instance, Mangolin et al. [162] have explored the multi-modality of acoustic and visual features. The proposed methodology

Figure 6.1: Proposed rhythm-based ADD methodology



Source: Elaborated by the author

is mainly composed of rhythmic features, however, we also consider visual features to verify the classification enhancement provided by a multi-modal system.

► **Rhythm-based Features:** they are a well-explored field of acoustic descriptors, extracting rhythmic properties from audio data, being initially explored in music genre classification. Over the years, their use was expanded to different tasks, e.g., movie genre classification [162], animal species [172], and age and gender identification [173]. The most prominent features in this field are presented as follows.

- *Rhythm Pattern (RP)* is an acoustic descriptor processed in two stages for each acoustic band. It is based on extracting sensitive frequencies to the human acoustic system based on rhythm [173]. The input audio signal is pre-processed through a single channel compression to calculate this descriptor, taking a consecutive group of six-second excerpts according to the current processing position. Next, each audio excerpt is conditioned to several transformations in the first stage, such as Fast Fourier Transform (FFT) and Bark Scale clustering, until the Sone representation. FFT is applied once again, followed by a weight modulation and a gradient filter in the second stage. Finally, a vector descriptor composed of 1440 values is generated based on the processed excerpts' median values;
- *Rhythm Histogram (RH)* is a rhythm-based acoustic descriptor in which the sum of acoustic bands produces a histogram of rhythmic energy [172]. The process is similar to RP until the Bark Scale step. After, the frequencies lower than 10 Hz

are summed up and allocated in a histogram. This descriptor generates a vector descriptor composed of 60 values, calculated through the average of each excerpt;

- *Statistical Spectrum Descriptor (SSD)* is an acoustic descriptor based on rhythm. Additionally, it has the potential to capture information related to different acoustic tones [173]. In a nutshell, the extraction process is also similar to RP. Various statistical moments are processed after applying the Bark Scale, such as mean, median, variance, skewness, kurtosis, min, and max value. As output, it is generated a vector descriptor composed of 168 values.

► **Visual Features:** Motivated by the results presented in Mangolin et al. [162], here work with Location Binary Patterns (LBP). It is a visual descriptor that seeks to summarize the pixel value patterns considering a given neighborhood. Thus, the descriptor starts from the premise that such patterns describe an image texture that can characterize patterns in images of the same class. Aiming to explore and evaluate the potential benefits of fusion among different descriptors (discussed in Section 6.4.2), in this study, we process an LBP descriptor, extracted from spectrum images of the audio signals, considering eight neighbors and two radii, which leads to a vector descriptor composed of 59 values.

6.4.2 Classification Model and Multimodality Definitions

We use as our classification model a DNN. It is a feed-forward artificial neural network with three hidden layers. It is important to mention that problems of drone identification will always need to deal with data imbalance [169]. This problem appears when the number of objects known for one class (e.g., drones) is much smaller than the number of objects in the second class (e.g., non-drones). For classifiers to reach good levels of accuracy in imbalanced problems, it is usually necessary to use sampling techniques, which can be undersampling the majority class or oversampling the minority class.

Fusion techniques consist of combining different descriptors and predictions to enhance the classification. These techniques can be grouped into two categories: early fusion and late fusion. Early fusion is based on the assumption that different descriptors can be combined to improve the classification of a given problem. This study intends to perform early fusion between acoustic and visual descriptors, using a specific classifier for this combination. Late fusion techniques, in turn, are based on the combination of predictions from different classifiers to address the final decision. This type of fusion can

Table 6.1: Early and late fusion configurations of the proposed rhythm-based strategy for ADD

<i>Descriptor / Configuration</i>	<i>Early Fusion</i>			<i>Late Fusion</i>			
	EF1	EF2	EF3	LF1	LF2	LF3	LF4
RH	✓	✓	✓	✓	✓		
RP		✓	✓				
SSD		✓	✓				
LBP	✓	✓					
EF1	-	-	-	✓	✓	✓	
EF2	-	-	-		✓		✓
EF3	-	-	-		✓	✓	✓

be performed using different combination rules, such as sum, product, maximum, and minimum [173].

Table 6.1 presents the considered configurations of early and late fusion. A check mark indicates the presence of the line descriptor (or configuration) in the column configuration. For instance, LBP is part of early fusion configurations EF1 and EF2. The result of EF1 classification, in turn, is applied as input for final classification in late fusion configurations LF1, LF2, and LF3. The hyphen mark means options that are not considered. In a nutshell, EF1 and EF2 attempt to improve our classification by combining visual and acoustic features. EF3, in turn, is a well-explored acoustic early fusion that commonly pointed out a high level of classification [173]. In the late fusion configurations, we combine and apply fusion rules considering the predictions provided by different early fusion configurations. Moreover, for LF1 and LF2, we also combine the prediction of RH descriptor solely, since different studies in the literature indicated that this descriptor could increase the identification rate in different contexts [162, 173].

6.4.3 Setup and Performance Evaluation

We implemented in Python version 3.5 the proposed methodology. We generated our classification models through the pre-built library provided by the scikit-learn¹ machine learning framework. The classification scripts and the data related to fold division and descriptors are available on a public website². This section describes the baseline dataset, the classification setup, and the description of the carried-out parameter tuning toward a better DNN classification model.

¹<https://scikit-learn.org/stable/>

²<https://sites.google.com/view/rhythmofdrones>

Table 6.2: Folds distribution regarding training and testing of the proposed rhythm-based strategy for ADD

Fold	Non-Drone	Bebop	Mambo
<i>Training Set</i>			
Validation 1	266	133	133
Validation 2	266	133	133
Validation 3	266	133	133
Validation 4	266	133	133
<i>Testing Set (Balanced)</i>			
Classification 1	266	133	133
<i>Testing Set (Non-balanced)</i>			
Classification 2	9308	133	133

► **Dataset:** We tested our methodology using a public drone acoustic signal dataset [169]. On this dataset, there are 10,372 audio samples characterized as “non-drone” and 1,332 audio samples from drones, divided into two models: “Bebop” and “Mambo”. Both used drone models are from the manufacturer Parrot³, belonging to the quadcopter type, i.e., they have four propellers and motors. In general, the Mambo drone has a shorter flight range, being about 10 m against Bebop’s 15 m. The speed reached by Bebop reaches 16 m/s while Mambo reaches only 8 m/s. In addition, Mambo is also physically smaller than Bebop. Although Mambo is a drone with fewer resources, it has a camera and gyroscope, differentiating itself from Bebop in that it does not have GPS.

The characteristics related to engine, speed, and weight can make a difference in each of these models’ sound, making it possible to identify them. Note that the number of non-drone audio samples is almost eight times greater than drone samples, causing an imbalance of data and hampering an equal distribution into folds. Al-Emadi et al. [169] pointed out that non-balanced classes can develop classification models with over-fitting. Hence, we use an under-sampling on the non-drone samples set to deal with this issue. However, if we consider real-world IoD environments, ADD classification models will handle many non-drone audio signals compared to drone audio signals.

► **Classification Setup:** In this study, we evaluate two types of classification: (i) binary, classifying the audio signal as a non-drone or drone; and (ii) multiclass, classifying the audio signal as a non-drone, “Mambo drone”, or “Bebop drone”. Furthermore, class imbalance of audio sample categories can challenge drone identification. Hence, we perform an undersampling of the majority class. The dataset keeps several non-drone equivalents to the number of drones for training, having a wholly balanced dataset.

To ensure diversity and statistical validation of our model, we apply the stratified k-fold cross-validation technique [173]. The audio data coming from the dataset is divided equally into k folds, consistently accounting for and preserving the class distribution.

³<https://www.parrot.com/en/drones>

Table 6.3: Final configuration of parameter tuning and the corresponding accuracy of the proposed rhythm-based strategy for ADD

Solv.	Actv	Learn.	Iter.	Hidden Size	Descrip.	Accur.	Stdv.
Adam	RELU	invscaling	500	(200,200,200)	SSD	0.9534	± 0.0070
					SSD	0.9915	± 0.0009
					RH	0.9967	± 0.0015

During the k -fold cross-validation process, $k - 1$ folds are considered to train and model and a fifth to validate the model learned. This process is repeated k times, and the results reported correspond to the average over all iterations. We use the $k = 4$ for tuning the values of the networks and use out-of-sample data to test the results, i.e., we consider a sample that was not used during the cross-validation procedure.

The out-of-sample testing is performed over two different scenarios: the balanced and unbalanced test sets (see Table 6.2). The unbalanced dataset is generated by returning the non-drone examples from the balanced dataset to the test set. Note that this simulates a scenario much closer to reality. Each fold has 266 non-drone samples, 133 Bebop, and 133 Mambo drone samples for training and parameter tuning validation. We perform the class balance similar to the seminal study [169] (50% for non-drone, 25% for Bebop, and 25% for Mambo). In the out-of-sample testing, the balanced dataset contains exactly the same number of samples, while in the unbalanced testing, the non-drone sample is increased from 266 to 9,308.

The validation folds (1 to 4) refer to the samples applied to train the DNN model following a 4-fold cross-validation technique for all classification scenarios: binary, multiclass, balanced, and non-balanced. Of course, for the binary classification, Bebop and Mambo are considered unique classes. The testing folds are discriminated in two classification sets: for the balanced scenario, the fold has 266 samples of the non-drone class; nonetheless, in the non-balanced scenario, there are 9,308 non-drone samples, embracing the ones of the balanced scenario including the remaining samples of the dataset.

► **Parameter Tuning:** Before testing the proposed methodology in out-of-sample data, we performed a parameter tuning of the DNN network. We use the 4-fold cross-validation to perform the parameter tuning according to the training and testing balanced set with binary classification. As a metric, we consider the DNN accuracy, which provides the pure success classification rate, considering the balanced dataset. The parameter tuning methodology is defined as follows: tuning (i) the hidden layer size; (ii) the maximum number of iterations; (iii) the learning rate; (iv) the activation function; and (v) the solver. In this tuning, we use *Approach 1* to perform the classification, considering the three rhythm-based descriptors.

Parameter tuning is an extensive procedure, demanding a series of evaluations and

Table 6.4: Binary classification results for balanced and unbalanced data regarding the performance evaluation of the proposed rhythm-based strategy for ADD

Input	Fusion Rule	Balanced Classes			Unbalanced Classes		
		Accuracy	F1-score		Accuracy	F1-score	
			Macro	Micro		Macro	Micro
Baseline[169]	-	0.9638 ± 0.0069	0.9590 ± 0.780	-	-	-	-
SSD	-	0.9426 ± 0.004	0.9426 ± 0.004	0.9426 ± 0.004	0.9557 ± 0.004	0.7598 ± 0.012	0.9557 ± 0.004
RP	-	0.9840 ± 0.004	0.9840 ± 0.004	0.9840 ± 0.004	0.9896 ± 0.001	0.9170 ± 0.004	0.9896 ± 0.001
RH	-	0.9981 ± 0.002	0.9981 ± 0.002	0.9981 ± 0.002	0.9969 ± 0.001	0.9733 ± 0.004	0.9969 ± 0.001
EF1	-	0.9971 ± 0.002	0.9971 ± 0.002	0.9971 ± 0.002	0.9977 ± 0.001	0.9800 ± 0.006	0.9977 ± 0.001
EF2	-	0.9736 ± 0.004	0.9736 ± 0.004	0.9736 ± 0.004	0.9792 ± 0.0028	0.8560 ± 0.013	0.9792 ± 0.002
EF3	-	0.9750 ± 0.004	0.9750 ± 0.004	0.9750 ± 0.004	0.9786 ± 0.0022	0.8526 ± 0.010	0.9786 ± 0.002
LF1	Max	0.9985 ± 0.002	0.9985 ± 0.002	0.9985 ± 0.002	0.9977 ± 0.001	0.9802 ± 0.004	0.9977 ± 0.001
LF1	Min	0.9985 ± 0.002	0.9985 ± 0.002	0.9985 ± 0.002	0.9977 ± 0.001	0.9802 ± 0.004	0.9977 ± 0.001
LF1	Sum	0.9985 ± 0.002	0.9985 ± 0.002	0.9985 ± 0.002	0.9977 ± 0.001	0.9802 ± 0.004	0.9977 ± 0.001
LF1	Product	0.9985 ± 0.002	0.9985 ± 0.002	0.9985 ± 0.002	0.9977 ± 0.001	0.9802 ± 0.004	0.9977 ± 0.001
LF2	Max	0.9938 ± 0.005	0.9938 ± 0.005	0.9938 ± 0.005	0.9958 ± 0.002	0.9642 ± 0.014	0.9958 ± 0.001
LF2	Min	0.9938 ± 0.005	0.9938 ± 0.005	0.9938 ± 0.005	0.9958 ± 0.002	0.9642 ± 0.014	0.9958 ± 0.001
LF2	Sum	0.9943 ± 0.004	0.9943 ± 0.004	0.9943 ± 0.004	0.9965 ± 0.001	0.9684 ± 0.010	0.9965 ± 0.001
LF2	Product	0.9948 ± 0.005	0.9948 ± 0.005	0.9948 ± 0.005	0.9963 ± 0.002	0.9684 ± 0.013	0.9963 ± 0.001

analyses. Due to the limited space, we present in Table 6.3 the final configuration of the carried parameter tuning and the accuracy results for the three descriptors. If we consider the study of Al-Emadi et al. [169], which is our baseline, the best accuracy rate for binary classification is 0.9638 ± 0.69 . It means that we have achieved better results than the baseline in this phase, considering SSD and RH descriptors solely. However, an out-of-sample experiment will be performed to test the generalization of the model.

6.4.4 Results and Discussion

This section discusses the results addressed by different descriptors and fusion techniques. All scenarios follow the classification models developed with the best parameters found during the tuning process, described in Table 6.3. We present binary and multiclass classification results, considering the balanced and non-balanced scenarios. We evaluate the classifications using two traditional metrics: accuracy and F1-score, to support our discussion.

The F1-score is the geometric mean of the precision and recall metrics and is appropriate when there is a class imbalance. For the multiclass scenario, We present both the macro and micro F1 measures. While the first is a non-weighted average for the F1-values for each class, the micro F1 calculates metrics globally by counting the total true positives, false negatives, and false positives, and hence accounts for class imbalance. The accuracy results presented are the average of the four classification models (one from each fold) created during tuning when applied to the out-of-sample data.

Table 6.5: Best confusion matrix of binary classification for balanced classes regarding the performance evaluation of the proposed rhythm-based strategy for ADD (LF1 – max rule).

Expected / Predicted	Non-Drone	Drone
Non-Drone	266	0
Drone	0	266

► **Binary Classification:** Here, we present and discuss the results of binary classification for balanced and non-balanced data. To make comparisons and discussions easier, they are summarized in Table 6.4. On the left side, we have the results of balanced data, and on the right side the ones of non-balanced data.

Regarding the classification considering the same number of samples per class, the results of LF1 –regardless of the fusion rule – reach the best classification accuracy compared to other variations. However, all descriptors and fusion rules obtain measures of accuracy and F1-score higher than 94%. If we do not consider the approaches with the SSD descriptor, the results are all higher than 97%, being better than the ones presented in Al-Emadi et al. [169]. Another essential aspect to note is that RH gives a high accuracy rate without using any fusion technique. This aspect also occurred in other related studies [172, 173], reinforcing the premise that RH catches sensible acoustic patterns based on the human auditory system.

Regarding unbalanced data, once again, LF1 reaches the best accuracy values considering all fusion rules. Except for SSD, the descriptors and fusion rules obtained accuracy values higher than 97%. Focusing on F1-Score, except for SSD, EF2, and EF3, the rates are always higher than 90%, showing the classifier captures the patterns of the drone class as well as the patterns from the non-drone class. Notice that we do not report the results for the baseline in this scenario, as the authors in [169] do not account for unbalanced cases, solving a simpler and less realistic problem.

Table 6.6 presents the best confusion matrix of one of the best results obtained, LF1 – max rule, as occurred in the classification scenario with the balanced dataset. Recall that the model that generated this result was chosen considering the best accuracy in the validation set. Observe that to perform the non-balanced binary classification, we did not retrain the classifiers but simply reproduced the out-of-sample data in a more realistic scenario, where 2.78% of the data belong to the drone class and the remainder to the non-drone class.

► **Multiclass Classification:** In this section, we present and discuss the results of multiclass classification for balanced and non-balanced data. In this case, the classification problem considers three classes: Non-Drone, Bebop, and Mambo, the latter two drone classes. The results are presented in Table 6.7. On the left side, we have the results of

Table 6.6: Best confusion matrix of binary classification with unbalanced classes regarding the performance evaluation of the proposed rhythm-based strategy for ADD (LF1 – max rule).

Expected / Predicted	Non-Drone	Drone
Non-Drone	9289	17
Drone	0	266

Table 6.7: Multiclass classification results for the balanced and unbalanced datasets regarding the performance evaluation of the proposed rhythm-based strategy for ADD

Input	Fusion Rule	Balanced Classes			Unbalanced Classes		
		Accuracy	F1-score		Accuracy	F1-score	
			Macro	Micro		Macro	Micro
Baseline [169]	-	0.9229 ± 0.012	0.9260 ± 1.320	-	-	-	-
SSD	-	0.9163 ± 0.012	0.9111 ± 0.014	0.9163 ± 0.012	0.9487 ± 0.0147	0.6879 ± 0.045	0.9487 ± 0.014
RP	-	0.9262 ± 0.002	0.9077 ± 0.002	0.9262 ± 0.002	0.9818 ± 0.0046	0.8084 ± 0.024	0.9818 ± 0.004
RH	-	0.9017 ± 0.006	0.9733 ± 0.008	0.9017 ± 0.006	0.9909 ± 0.0002	0.8394 ± 0.023	0.9909 ± 0.002
EF1	-	0.9055 ± 0.014	0.8776 ± 0.013	0.9055 ± 0.014	0.9861 ± 0.0122	0.8206 ± 0.079	0.9861 ± 0.012
EF2	-	0.9459 ± 0.011	0.9394 ± 0.011	0.9459 ± 0.010	0.9594 ± 0.0123	0.7435 ± 0.037	0.9594 ± 0.012
EF3	-	0.9426 ± 0.009	0.9351 ± 0.009	0.9426 ± 0.009	0.9657 ± 0.0196	0.7585 ± 0.073	0.9657 ± 0.019
LF3	Max	0.9539 ± 0.002	0.9417 ± 0.004	0.9539 ± 0.001	0.9984 ± 0.0040	0.9078 ± 0.036	0.9934 ± 0.004
LF3	Min	0.9591 ± 0.005	0.9480 ± 0.007	0.9591 ± 0.005	0.9935 ± 0.0046	0.9131 ± 0.035	0.9936 ± 0.004
LF3	Sum	0.9558 ± 0.002	0.9443 ± 0.004	0.9558 ± 0.002	0.9936 ± 0.0046	0.9108 ± 0.036	0.9936 ± 0.004
LF3	Product	0.9591 ± 0.004	0.9480 ± 0.006	0.9591 ± 0.003	0.9937 ± 0.0047	0.9137 ± 0.036	0.9937 ± 0.004
LF4	Max	0.9501 ± 0.007	0.9437 ± 0.009	0.9501 ± 0.007	0.9675 ± 0.0079	0.7635 ± 0.036	0.9675 ± 0.007
LF4	Min	0.9501 ± 0.007	0.9434 ± 0.009	0.9501 ± 0.007	0.9674 ± 0.0079	0.7640 ± 0.034	0.9674 ± 0.007
LF4	Sum	0.9501 ± 0.007	0.9437 ± 0.009	0.9501 ± 0.007	0.9674 ± 0.0080	0.7634 ± 0.036	0.9674 ± 0.008
LF4	Product	0.9501 ± 0.007	0.9436 ± 0.009	0.9501 ± 0.007	0.9675 ± 0.0078	0.7647 ± 0.033	0.9675 ± 0.007

balanced data, and on the right side, unbalanced data.

Regarding balanced data, although the late fusion technique LF3 presents the best results in binary classification, this approach significantly improves the accuracy rate in the multiclass scenario compared to early fusion and descriptors solely. In this scenario, LF3 with *min* and *product* rules reach the best accuracy, with values higher than 95%. Furthermore, observing all fusion rules applied with late fusion approaches (LF3 and LF4), we note that the accuracy is always higher than 95%, reinforcing that the late fusion combination of rhythm-based descriptors is a suitable approach to multiclass classification in this dataset.

Table 6.8 shows the confusion matrix obtained by the classification model with the best accuracy results in the validation set, LF3 - Product Rule. All classes have high precision, mainly in the identification task of non-drones, with 99.62%. Besides, the identification of drones also has high precision, with fewer mistakes for the Mambo class (1.87% of misclassifications). We also note that it is more difficult for the classification model to identify the Bebop drone correctly.

Regarding unbalanced data, LF3 with *Sum* rule obtains the best accuracy: 0.9984. The other LF3 rules also reach an accuracy rate greater than 99%. However, given the class unbalance, in this scenario the f-measure is a more appropriate measure to evaluate. For both Macro and Micro F1-scores, LF3 also presents the highest rates, with rates greater than 90% (Macro) and 99% (Micro). We also can note that the remaining F1

Table 6.8: Confusion matrix for multiclass classification with balanced classes regarding the performance evaluation of the proposed rhythm-based strategy for ADD (LF3 – product rule).

Expected / Predicted	Non-Drone	Bebop	Mambo
Non-Drone	265	0	1
Bebop	0	119	14
Mambo	1	4	128

Table 6.9: Confusion matrix for multiclass classification of unbalanced data regarding the performance evaluation of the proposed rhythm-based strategy for ADD (LF3 – max rule).

Expected / Predicted	Non-Drone	Bebop	Mambo
Non-Drone	9293	2	11
Bebop	0	128	5
Mambo	6	13	114

Macros address rates are lower than 85%. This aspect can be explained due to the fact that this metric does not take into account the imbalanced scenario. This issue is solved using F1 Micro, which considers data imbalance. Table 6.9 shows the confusion matrix of the LF3 – max rule. Again, the classification models trained with the balanced data were reused in the out-of-sample data, which included more non-drone instances to simulate a real scenario. Here, the class distribution was 97.22% for non-drone, 1.39% for Bebop, and 1.39% for Mambo. Once more, in the unbalanced classification scenario, only the number of instances of the non-drone class increased. Thus, when comparing the confusion matrices of the best balanced (Table 6.8) and non-balanced (Table 6.9) results, it appears that the errors refer to classifying instances of the type non-drone erroneously as a drone.

► **Overall Comparison:** Table 6.10 shows the best accuracy results of Al-Emadi et al. [169] and our study. We use only the balanced results for a fair comparison since Al-Emadi et al. [169] do not present results for non-balanced scenarios. As discussed, we perform the class balance in the same way as this baseline study. Table 6.11 shows the comparison between the two best results from our study and Al-Emadi et al. [169] considering the macro F1-score. Once again, we present better results of the f-score and with a lower variance. As we can note, there is an enhancement in terms of accuracy for both binary and multiclass classification, being 3.47% and 2.97%, respectively. Note that in classification problems, the higher the accuracy, the more challenging to improve classification results. Hence, this is indeed a significant improvement. Also, note that the standard deviation results are very low, also showing statistical validity in our findings.

Table 6.10: Comparison of the baseline and our proposal – Best accuracy rate for binary and multiclass classification in balanced data scenarios.

Input	Al-Emadi et al. [169]	Our Study
Binary	0.9638 ± 0.0069	0.9985 ± 0.0024
Multiclass	0.9294 ± 0.0118	0.9591 ± 0.0038

Table 6.11: Comparison of accuracy and F1-score between the baseline and our proposal.

Input	Accuracy	F1-Score
Al-Emadi et al. [169]	0.9294 ± 0.0118	0.9263 ± 0.0132
Our Study (LF3 - Min)	0.9591 ± 0.0050	0.9480 ± 0.0070
Our Study (LF3 - Prod.)	0.9591 ± 0.0038	0.9480 ± 0.0060

► **General Discussion:** In a nutshell, rhythmic-based descriptors allied with fusion techniques empower ADD, useful in different tasks, including those related to malicious attacks. Our classification model improved 3.47% the baseline binary classification, in which we addressed 0.9985 of accuracy rate using the early fusion technique. For multiclass classification, we improved the baseline model by 2.97%, reaching an accuracy of 0.9591 through the late fusion technique with *min* and *product* rules. Furthermore, considering that in real-world drone-based networks, ADD tasks will handle a higher number of non-drone acoustic signals, we carried out experiments in unbalanced data scenarios. We addressed accuracy and Macro F1-score of 98.02% with late fusion techniques in binary classification. In multiclass classification, the model addressed a Macro F1-score of 91.37%.

6.5 Design of DissIdent

DissIdent is a dissimilarity-based and distributed approach for identifying unknown UAVs in the airspace. The approach works as a distributed mechanism, where the cooperation between drones and ZSPs is a key-enabling task to address high-level detection. Two subsequent tasks compose the architecture of DissIdent: the deployment and monitoring phases. The deployment phase is supported by learning models, while both learning and clustering techniques embrace the monitoring phase. Before presenting this architecture, we discuss the dissimilarity concepts and how to transform an input signal from the airspace to the dissimilarity space, which is a fundamental concept in our proposal.

6.5.1 Dissimilarity Representation

Dissimilarity refers to a function of proximity between two observed samples [164], transitioning from the feature set to the dissimilarity set. In this latter, each element represents a distance between the samples instead of the processed feature value. One of the main advantages of the dissimilarity approach is to reduce a very large classification problem to a binary problem. This technique emerged as a suitable solution to address both large multi and independent-class problems, mapping them to binary problems through dichotomy transformations [164]. Compared to feature-based classifiers [92, 174], it mitigates the trade-off between the tractability and accuracy of large multi-class problems, providing similar results at a reasonable processing time, mainly in clustering tasks.

Let us consider a set of IoD signal inputs \mathcal{SI} and a set of feature vectors \mathcal{FV} . Also, let us consider a feature vector $\mathcal{FV}_i \in \mathcal{FV}$, $i \in \mathcal{SI}$, representing the data processed from \mathcal{SI} following a given descriptor d . Let Euc_{ij} be the dissimilarity vector between two feature vectors $\mathcal{FV}_i, \mathcal{FV}_j$, $i, j \in \mathcal{SI}$, calculated according to Eq. 6.1. Euc_{ij} represents the Euclidean distance between the feature vectors, denoting a transformation from the feature to the dissimilarity space [164].

$$Euc_{ij} = |\mathcal{FV}_i[k] - \mathcal{FV}_j[k]| \forall k \in \mathcal{FV}_i \quad (6.1)$$

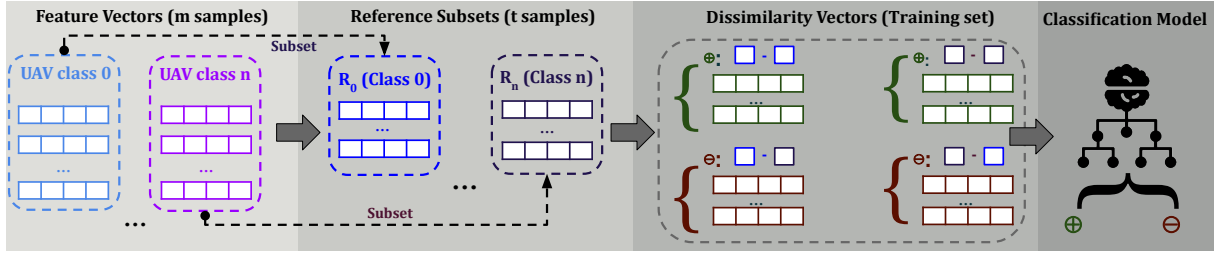
Euc_{ij} is a suitable input to a supervised-based model because it keeps a concise relation between any two feature vectors, represented with the same dimension of these features. Hence, it must have an associated dissimilarity class $C_d(Euc_{ij})$: x_{\oplus} or x_{\ominus} , defined according to Eq. 6.2, where C_f represents the class regarding the feature space.

$$C_d(Euc_{ij}) = \begin{cases} x_{\oplus} & \text{if } C_f(\mathcal{FV}_i) = C_f(\mathcal{FV}_j) \\ x_{\ominus} & \text{otherwise} \end{cases} \quad (6.2)$$

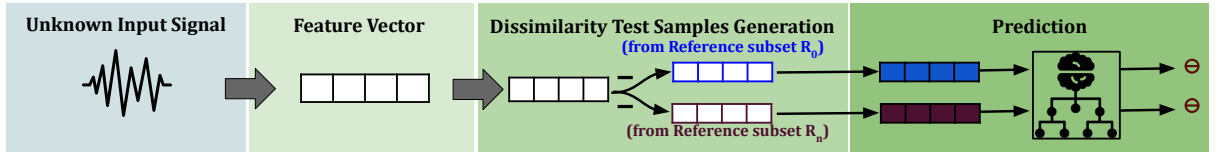
Dissimilarity classes apply to binary classification models in the training and testing steps. They are discussed below.

► **Dissimilarity for Training Models:** Dissimilarity classes represent a classification model's input data for well-known input signals. Fig. 6.2a shows the workflow of how to use dissimilarity towards training it. From the feature vectors, we generate reference subsets whose elements are combined to create dissimilarity vectors. These vectors are the input for training the classification model. An important aspect is that the generation of dissimilarity vectors follows a combinatorial order of the number of feature vectors from different feature classes. However, the input training size can be unfeasible for n sufficiently large. Thus, we consider a maximum value t of samples to deploy n reference subsets $R_1, \dots, R_n \subset \mathcal{FV}$.

Figure 6.2: Dissimilarity representation workflow



(a) Workflow for available data. This workflow proposes the training of classification models



(b) Workflow for unknown data. This workflow depicts the process of predicting unknown signals in the environment

Source: Elaborated by the author

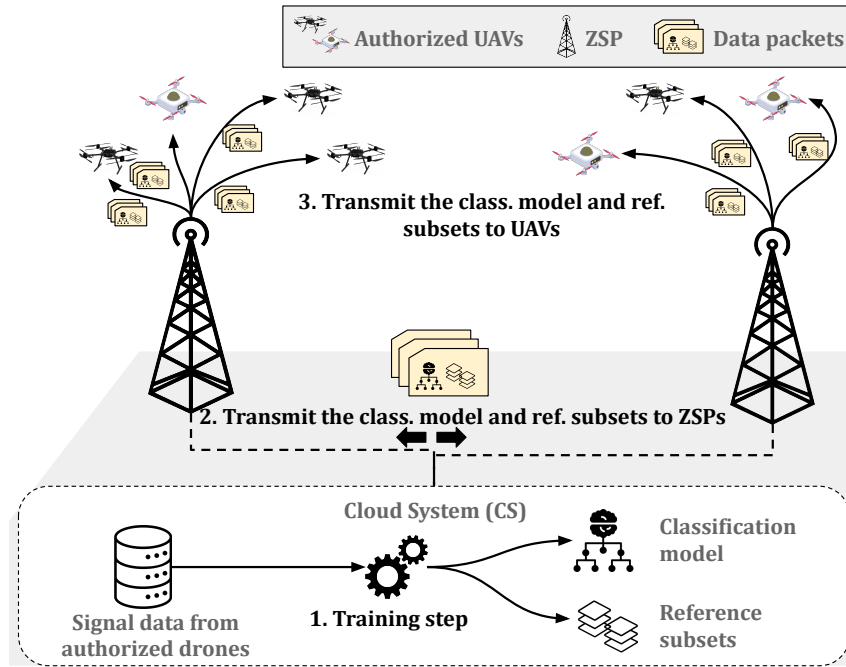
We compute the training set T providing a balanced number of positive and negative samples as follows: we process the dissimilarity of feature vectors from the same R , resulting in $\binom{\tau}{2}$ samples of x_{\oplus} ; next we generate the same number of x_{\ominus} samples computing the dissimilarity between each feature vector of R with a vector from another reference subset, randomly chosen. It gives a training set T with $2n\binom{\tau}{2}$ samples.

► **Classification Model Representation:** Fig. 6.2b depicts the dissimilarity used to detect unauthorized UAVs. Let us consider a feature vector \mathcal{FV}_u generated from a signal to be classified. In this case, the main goal is to know if \mathcal{FV}_u is similar to some known category set or dissimilar to all, representing a new category in this system. To address this goal, we generate n different dissimilarity vectors, computing the dissimilarity between \mathcal{FV}_u and one feature vector (randomly chosen) from each reference subset R_1, \dots, R_n . Next, we submit these vectors to \mathcal{M} , getting the binary classification.

The prediction output of each dissimilarity vector provides information about the unknown input signal. If the predictions for all vectors are x_{\ominus} , then the input signal corresponds to an unknown class. In this case, the model creates a new subset R_u with \mathcal{FV}_u as an element. On the other hand, if the classifier predicts one or more dissimilarity vectors as x_{\oplus} , the input signal comes from a known entity in the network. In this case, the class $C_f(\mathcal{FV}_u)$ will correspond to the same class of the paired vector of R from the dissimilarity vector with x_{\oplus} classification with higher prediction.

Considering the airspace, both the propeller's sound and the RF-based signals from authorized drones can serve as input signals for training dissimilarity models. Thus, robust classification models can be deployed *a priori* in the authorized UAVs to perform the identification cooperatively. Apart from that, the environmental signals (coming from

Figure 6.3: DissIdent approach: Deployment Phase



Source: Elaborated by the author

both authorized and unknown sources) can be submitted to these classification models, and then, a global cloud system can decide the presence of an unknown drone.

6.5.2 DissIdent Deployment Phase

Figure 6.3 illustrates the DissIdent architecture regarding the Deployment Phase. Before starting to perform the identification tasks, the authorized drones need a trained classification model based on their data. Hence, the deployment phase must follow the dissimilarity representation workflow.

Therefore, to deploy the classification model, DissIdent requires the feature vectors $\mathcal{FV}_1, \dots, \mathcal{FV}_n$ from n authorized entities in the environment. We consider both acoustic and RF-based features in this study. From pre-collected data of the authorized drone, a centralized Cloud System (CS) can generate the reference subsets $\mathcal{R}_0, \dots, \mathcal{R}_n$, and the dissimilarity vectors. Hence, a classification model can be trained following some well-defined learning model, such as Multi-Layer Perceptron (MLP) and Support-Vector Machines (SVM). Nonetheless, DissIdent is not dependent nor limited by any feature vector representation or learning techniques, and different arrangements can be considered. We assume that this training step is an offline task, occurring before the beginning of a DaaS application.

With the trained classification model, the CS transmits both the models and the reference subsets to the ZSPs. They transmit these data to the drones in their controlling region through the communication channel. After that, the IoD nodes can start to identify the potential intruders cooperatively.

6.5.3 Monitoring Phase

Authorized drones and the ZSPs perform the monitoring of the environment. In this phase, we assume that a given drone constantly beaconing in the network, updating its location to a ZSP [5]. Thus, in this beaconing, the UAV can inform when it detects an unknown signal. With the data from the ZSPs and drones, the CS processes and makes the final decision about the presence of an unknown drone. This phase involves two main tasks, described in the following.

► **Node’s Monitoring:** The node’s monitoring follows the dissimilarity workflow for unknown data (previously presented in Fig. 6.2b). Initially, a given node generates a feature vector from the input signal. After, for each reference subset, DissIdent processes the dissimilarity test samples, being classified by a pre-trained model. If the classification of at least one reference set addresses a greater positive prediction rate, the input signal is associated with the class with the higher positive prediction rate. If this class is related to samples from an unknown entity detected previously, a warning is sent to the CS considering a time interval Δ_t . Similarly, if no reference set addresses a positive prediction rate, the signal can come from a new unknown entity, and the CS must also be warned. However, the warning is sent immediately since it can represent an entity not yet identified.

► **Decision Protocol:** Since the classification models can not provide 100% efficacy, the final decision about detecting an unknown UAV must consider as many nodes as possible. Hence, DissIdent has a CS decision protocol (Algorithm 17) that gathers the warnings transmitted by the nodes during a given time interval, evaluating them into clusters to make a final decision.

Algorithm 17 formally describes the decision protocol. It requires as input the transmitted warning set $WARN$, the reference sets R , and two numerical values ρ and τ , indicating a distance radius and a number of elements, respectively. Initially, the warnings are grouped according to the ZSPs that transmit them (Line 1). For each group, the warnings are also separated into two groups: the ones related to the detection

of new unknown signals ($WARN_{\ominus}$) (Line 4) and the ones associated with the detection of an existent cluster ($WARN_{\oplus}$) (Line 5).

After, DissIdent applies the decision over these groups. The decision process is presented as a distinct function for better visualization (Lines 13–27). Given the group of identifications, the first step is to filter the warnings from drone nodes (Line 14). If there is no warning coming from drone nodes (Lines 15–20), the warning comes from a ZSP node. In this case, it is necessary to verify if there are at least N_{near} drones near the ZSP, considering a distance radius ρ that could also identify the potential unknown signal (Line 16). The detection of an unknown signal is a false positive if there are not enough drones (Lines 18–19). Otherwise, the unknown drone presence is confirmed (Line 17).

On the other hand, if there are drones that warn about a potential unknown signal (Lines 21–27), DissIdent establishes a 3-dimensional perimeter considering the position of these drones (Line 22), verifying the number of drones N_{inside} inside this perimeter (Line 23). The idea is: if there are drones inside the region and they do not warn about the unknown signal, the warning can be a false positive. If more drones detect the issue than N_{inside} (Line 24), then it is confirmed. Otherwise, it is taken as a false positive.

Based on the decisions, the algorithm takes different actions regarding $WARN_{\ominus}$ and $WARN_{\oplus}$. If the identification of a new unknown entity ($WARN_{\ominus}$ group) is confirmed, DissIdent creates a new reference set R_{new} containing the feature vectors related to this group, representing a new cluster. This information is sent to all the nodes (Lines 7–8). For $WARN_{\oplus}$, the related reference set of the existent unknown drone is updated with the provided sample features (Lines 10–11).

► *Time Complexity Analysis:* The grouping of warning messages (Line 1) occurs in a linear time $\Theta(|WARN|)$ using set operations. The decision protocol occurs for each group of warnings, constrained in the worst case by the number of ZSPs in the network $\mathcal{O}(|ZSPs|)$ (loop of Lines 2–11). The gathering of drones' updated information (Line 3) has a time complexity of the number of drones, in the worst case $\mathcal{O}(|\mathcal{D}|)$. The deployment of sets $WARN_{\oplus}$ and $WARN_{\ominus}$ (Lines 4–5) have both a linear time complexity based on the number of elements of $WARN_{zsp}$. The subsequent steps (Lines 6–11) depend on the **Decision** function. In a nutshell, the time complexity of this function is dominated by operations that iterate over the total number of drones in the worst case. Thus, **Decision** has a time complexity of $\mathcal{O}(|\mathcal{D}|)$. Considering that the size of $WARN$ depends on the number of ZSPs and that the size of $WARN_{zsp}$ is constrained by the number of drones, the time complexity of Algorithm 17 is related to the number of ZSPs and drones directly, as presented by Equation 6.3.

$$\begin{aligned} T(\text{Decision-Protocol}) &= \Theta(|WARN|) + \mathcal{O}(|ZSPs|(|\mathcal{D}| + |WARN_{zsp}|)) \\ &= \mathcal{O}(|ZSPs||\mathcal{D}|) \end{aligned} \quad (6.3)$$

Algorithm 17: Decision-Protocol

```

Input :  $WARN, R, \rho, \tau$ 
1  $WARN_{group} \leftarrow$  group the elements of  $WARN$  by the ZSPs
2 foreach  $WARN_{zsp}$  of  $WARN_{group}$  do
3    $D_{zsp} \leftarrow$  get the updated information of the drones flying over the  $zsp$ 's airspace
4    $WARN_{\ominus} \leftarrow$  get the elements of  $WARN_{zsp}$  pointed as unknown entities
5    $WARN_{\oplus} \leftarrow$  get the elements of  $WARN_{zsp}$  pointed as belonging to an existent
   cluster
6   if  $Decision(WARN_{\ominus}, D_{zsp}, \rho, \tau)$  then
7      $R_{new} \leftarrow$  create a new reference set for the unknown signal, with the feature
     vectors provided by  $WARN_{\ominus}$ 
8     send  $R_{new}$  to all authorized nodes
9   if  $Decision(WARN_{\oplus}, D_{zsp}, \rho, \tau)$  then
10     $R_i \leftarrow$  get the reference set from  $R$  related to the existent cluster of the element
    in  $WARN_{\oplus}$ 
11    Update  $R_i$  with the feature samples provided by  $WARN_{\oplus}$ 
12 Function  $Decision(W, D, \rho, \tau)$ :
13    $D_{warn} \leftarrow$  filter the elements of  $W$  that are drones
14   if  $D_{warn} = \emptyset$  then
15      $N_{near} \leftarrow$  calculate the number of drones of  $D_{zsp}$  near to  $zsp$  considering a
     radius  $\rho$ 
16     if  $N_{near} \leq \tau$  then
17       return true
18     else
19       return false
20   else
21      $Perim_{warn} \leftarrow$  process a 3-dimensional perimeter considering the drones of
      $D_{warn}$ 
22      $N_{inside} \leftarrow$  calculate no. of drones of  $D_{zsp}$  inside  $Perim_{warn}$ 
23     if  $N_{inside} \leq |D_{warn}|$  then
24       return true
25     else
26       return false

```

6.5.4 Simulation Setup and Performance Evaluation

Using IoDSim, we simulate a robust IoD network infrastructure [5] in a region of Manhattan Island, NY, where fifty authorized drones from five different models (Parrot AR 2.0, Parrot Mambo, DJI Inspiron 1 Pro, DJI Matrice 100, and DJI Matrice 600) fly and monitor the environment for thirty minutes. We consider eight non-authorized drones from two models (DJI Phantom 3 and DJI Phantom 4) representing the unknown entities. Twenty-four different ZSPs manage the airspace and monitor the environment.

Regarding the input signals, we consider acoustic and RF-based samples from the five types of authorized drones and the two unauthorized types. These data come from

three different datasets: (i) MPACT RF-based dataset [175]; (ii) DroneRF [176]; and (iii) the UAV acoustic-based dataset [169]. During the simulation experiments, these data are linked with the related drones.

► **Features and Classification Models:** Regarding the acoustic-based signals, we apply the use of rhythmic-based features (presented in Section 6.4). As previously discussed, these features can provide an empowered representation of the drones’ sound, providing higher identification rates. For the RF-based signals, we extract fifteen statistical features (e.g., kurtosis and entropy) [177]. The generation of these features from the input signals follows the guidelines described in these studies, adopted here as baselines [174, 177]. From the features, we generate the related dissimilarity vectors and, consequently, the training set for the classification models, as depicted in Fig. 6.2a.

The deployed classification models also follow strategies designed in the baseline studies. Thus, our experiments have two distinct classification models. Regarding the acoustic signals, we use the DNN modeled in Section 6.4, also conducting the referred parameter tuning, but with the features in the dissimilarity space. Our final parameterization addresses the same configuration as presented in Section 6.4. Likewise, we use an SVM model to classify the RF-based signals [177].

► **Compared Approaches:** We analyze the robustness of DissIdent comparing its performance with supervised-based techniques and clustering strategies. Regarding the first group, we consider a DNN-based UAV detection model designed specifically for acoustic-based signals considering rhythm-based features, and an SVM model [177] designed to detect drones from RF signals. As clustering models, we consider the three strategies discussed previously: DBSCAN, OPTICS, and BIRCH. We apply these clustering models for both acoustic and RF-based signals, distinctively. Hence, we compare DissIdent with eight different models, labeled as follows: DNN, SVM, DBSCAN-AC, DBSCAN-RF, OPTICS-AC, OPTICS-RF, BIRCH-AC, and BIRCH-RF. For the clustering approaches, labels ending with “AC” and “RF” refer to the use of acoustic and RF signals as data input, respectively.

We present in the following the general information regarding the configuration of each evaluated approach.

- *DissIdent setup:* Any authorized nodes in the IoD act as monitoring actors to DissIdent. As the sound of the drone’s propellers can cause a very loud noise in the system (without the possibility of filtering), drones are responsible for monitoring only the RF signals. ZSPs, in turn, perform these tasks for both acoustic and RF signals. Hence, this setup highlights DissIdent as a multi-modal approach. Aiming to investigate the impact of the decision protocol of DissIdent, we configure the

Table 6.12: Summarization of the results regarding the accuracy obtained by each approach

Approach	Signal Input		Accuracy of Known Signals (%)						Accuracy of Unknown Signals (%)		
	Acoust.	RF	General	AR	Beb	Insp1	Mat100	Mat600	General	Phant3	Phant4
SVM		✓	98.84±0.2	91.35±0.8	93.86±0.7	95.95±0.9	95.38±0.8	93.24±0.5	90.33±0.4	-	-
DNN	✓		98.26±0.4	92.28±0.4	95.20±0.3	90.07±0.4	93.20±0.3	90.40±0.4	92.62±0.9	-	-
DBSCAN-AC	✓		97.69±0.9	93.57±1.3	89.52±1.0	90.84±1.2	81.72±1.8	95.58±1.2	81.71±1.9	80.51±1.1	78.26±2.5
DBSCAN-RF		✓	97.56±0.9	92.55±1.4	87.23±1.2	91.22±1.1	78.54±1.1	94.40±1.4	79.84±1.9	79.30±2.0	74.84±2.3
OPTICS-AC	✓		96.37±1.3	90.80±2.1	91.94±2.2	93.89±1.9	92.92±1.5	88.18±1.9	79.49±1.9	76.84±1.9	77.57±1.8
OPTICS-RF		✓	96.46±0.9	87.84±1.0	91.26±1.2	94.06±1.0	92.17±1.4	78.35±1.4	77.78±2.5	72.75±2.9	76.22±2.8
BIRCH-AC	✓		95.76±1.3	88.87±0.3	81.64±0.9	81.70±0.1	77.89±1.8	84.40±0.9	79.57±0.5	75.72±0.9	76.89±1.1
BIRCH-RF		✓	95.90±2.1	87.30±0.3	79.72±0.9	77.98±0.3	76.59±1.4	85.89±0.8	76.78±2.3	69.48±4.3	77.49±1.9
DissId-1	✓	✓	99.60±0.2	91.89±1.8	90.30±1.3	90.87±1.2	93.20±0.9	94.38±1.1	87.63±0.1	87.66±0.1	85.46±0.4
DissId-60	✓	✓	99.63±0.3	94.05±0.2	91.71±1.1	91.52±1.5	94.31±1.4	94.93±0.9	94.69±0.6	93.24±0.7	93.02±0.8

approach with two different time intervals Δ_t : 1 second, and 60 seconds. They are labeled as DissId-1 and DissId-60, respectively;

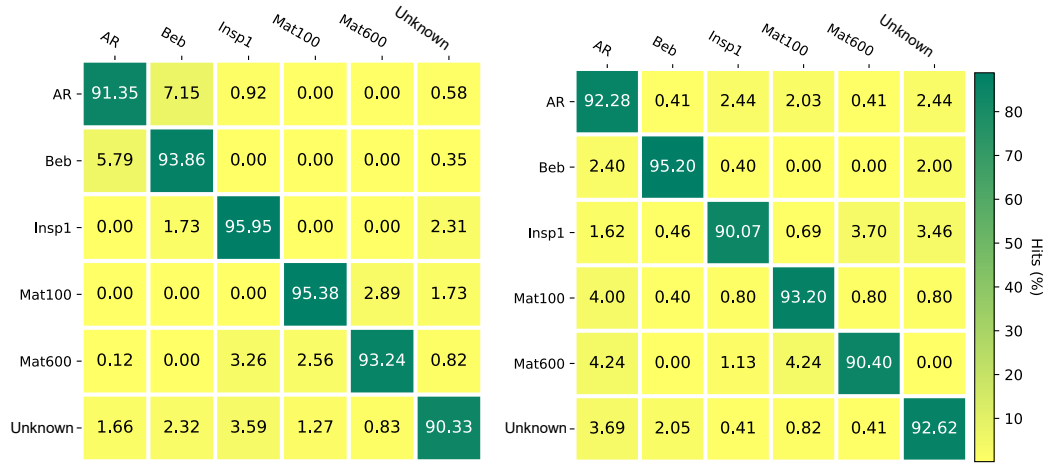
- *Supervised-based techniques setup*: We tune both the SVM and DNN models following the parameter tuning described in the referred related studies [174, 177]. The models were trained considering distinct categories for each authorized drone model, and one category for unknown signals, totaling six categories;
- *Clustering models setup*: As discussed, the related clustering models have no awareness of the categorization of the input signal in terms of each drone model. Therefore, we tune each clustering model with input data from authorized drones, representing a model defined *a priori*, following a process similar to the DissIdent Deployment Phase (Section 6.5.2). As the considered feature descriptors build a signal representation with a considerable range of values, we perform a data dimensional reduction, mapping the input feature into a three-dimensional representation.

6.5.5 Results and Discussion

This section presents the results of the experimental evaluation. They correspond to the average of 30 simulations for each approach, with 95% interval confidence. We divide the discussion in (a) an analysis regarding the general detection of known and unknown signals; and (b) an analysis regarding the effective identification of signals, focusing on the distinction between different UAV models from the unknown signals.

Table 6.12 supports our analysis, summarizing the accuracy regarding both UAV detection and identification. Results marked in bold with a dark gray background correspond to the best accuracy average. Results with a light gray background also correspond to the best accuracy but considering the interval error. Also, we detail these results in Figures 6.4, 6.5, and 6.6, where confusion matrices indicate the average hit rate for each

Figure 6.4: Confusion Matrix regarding the results of supervised-based approaches for UAV identification



(a) SVM model for RF signals

(b) DNN model for acoustic signals

Source: Elaborated by the author

Figure 6.5: Confusion Matrix regarding the results of DissIdent UAV identification for each configuration

(a) DissIdent Identification, $\Delta_t = 1$ sec(b) DissIdent Identification, $\Delta_t = 60$ secs

Source: Elaborated by the author

pairwise of UAV models. Given that the supervised models can not distinguish the two unknown models, they classify the related signal inputs as “unknown”. For the Clustering and DissIdent approaches, besides the referred unknown drones’ models, the matrices have a mismatching column. It occurs when the approach identifies an authorized node as unknown or when an additional cluster is generated, fragmenting the clusters and increasing their expected amount.

► **UAV Detection Analysis:** DissIdent overcame all the compared approaches regarding the detection task, as highlighted in Table 6.12 specifically in both columns labeled as “General”. Although the approaches addressed a high accuracy in identifying known signals (all of them with averages higher than 95%) DissIdent configurations addressed

an average of about 99% of correct detection.

Observing the detection of unknown signals, DissIdent with $\Delta_t = 60$ sec addressed almost 95% of accuracy, while the supervised-based models addressed about 90 and 92%. On the other hand, clustering approaches did not address proper rates of detection for unknown signals, in which the best average accuracy was about 81% for DBSCAN with acoustic signals. These low accuracy rates can be caused due to the data dimensional reduction. However, a thorough analysis must be carried out to evaluate its influence.

Furthermore, these results pointed out two important insights about DissIdent. Firstly, the transformation from the feature to the dissimilarity space did not cause a loss of information for the training/classification model as well for the deployment of the reference subsets. Indeed, they promoted a slight enhancement in the detection of known signals and a significant improvement regarding unknown signals detection. Secondly, the DissIdent approach whose time interval Δ_t is 60 seconds showed up as a better configuration to detect unknown signals. This result indicates a better deployment of the reference subsets, being able to gather more environmental information before building and releasing the subsets to the drones.

► **UAV Identification Analysis:** Here, we analyze the performance of the approaches regarding the correct identification of drone models, considering the five authorized drone models and the two unauthorized ones. As the supervised-based models can not distinguish the two unauthorized models, the discussion involving the identification of unknown signals refers to the clustering and DissIdent approaches, only.

As occurred in the detection task, DissIdent approaches addressed identification rates higher than 90% for all the authorized models. However, the supervised-based models also addressed high rates, overcoming DissIdent in some situations. For instance, the SVM model addressed an accuracy higher than 95% for Inspiron1 and Mat100 models while DissIdent with $\Delta_t = 60$ sec addressed an accuracy of about 91.5 and 94.3, respectively. Nonetheless, considering the interval error, the results are closer.

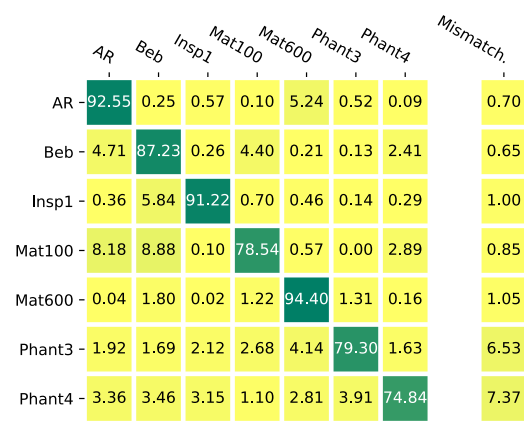
Observing the confusion matrices presented in Figures 6.4 and 6.5, we can note that the supervised-based models have a slight confusion in the AR, Bebop, and Inspiron1 models and almost no confusion for the remained ones. On the other hand, DissIdent wrongly classified the models more evenly, yet still slightly. This characteristic turned over between authorized models and the unknown signals. In this case, DissIdent approaches had almost no misclassifications while the supervised-based models addressed a slight confusion.

Clustering approaches, in turn, presented a higher confusion between both authorized models and unknown signals. as presented in Figure 6.6. Regarding the classification of the unknown signals, we can observe that DBSCAN and OPTICS performed a mismatching higher than 10% for acoustic signals and higher than 6% for RF signals. On

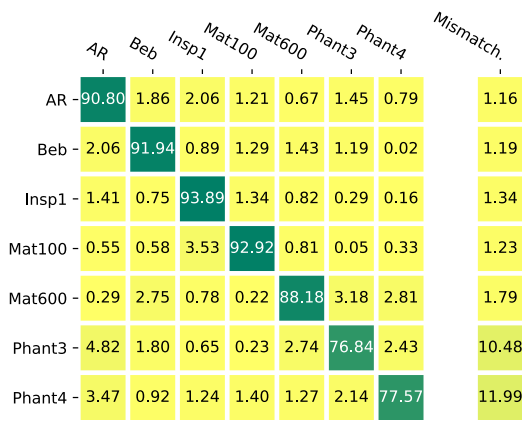
Figure 6.6: Confusion Matrix regarding the results of the Clustering approaches



(a) DBSCAN for acoustic signals



(b) DBSCAN for RF signals



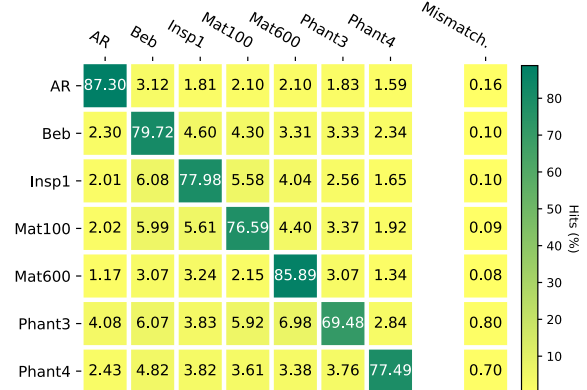
(c) OPTICS for acoustic signals



(d) OPTICS for RF signals



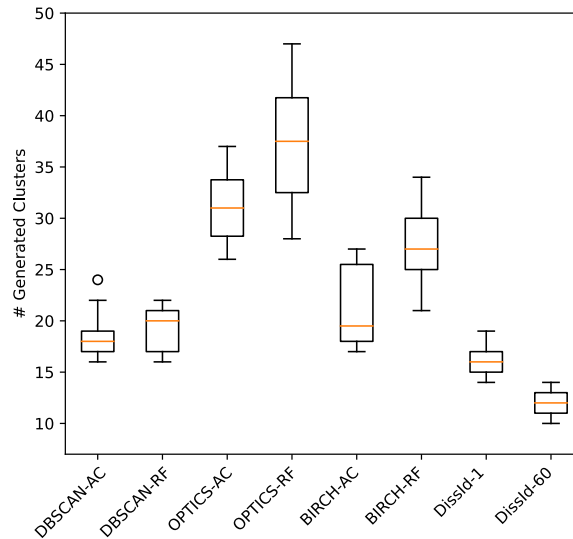
(e) BIRCH for acoustic signals



(f) BIRCH for RF signals

Source: Elaborated by the author

Figure 6.7: Results of clusters distribution for each clustering technique



Source: Elaborated by the author

the other hand, BIRCH had a minor mismatching, but a higher confusion with authorized nodes. Nonetheless, DissIdent overcame all the clustering approaches regarding the identification task, mainly for the unknown signals where the generated confusion was significantly lower, leading to better accuracy.

The number of generated clusters can better explain the lower performance of clustering approaches. Figure 6.7 presents this information considering all the experiments. Compared to DissIdent, clustering approaches produced a higher number of clusters, mainly OPTICS and BIRCH. Furthermore, these approaches had a significant variance over the simulations while DissIdent kept concise. Considering that the expected number of clusters was 7 (one for each drone model), DissIdent with $\Delta_t = 60$ seconds increased this number the double at most, while some clustering approaches produced six times more clusters, such as OPTICS.

Hence, we can observe that the high number of clusters leads to the fragmentation of the identified data, hampering the accuracy rate. As discussed in the UAV detection analysis, the deployment of the reference subsets plays a fundamental role in addressing significant rates of identification for unknown signals, reinforcing the robustness of DissIdent.

6.6 Chapter Remarks

In this chapter, we introduced new research fronts related to ADD strategies, proposing two new techniques: a smart strategy to detect drones using rhythmic-based features, extracted from the drone’s propeller sound; and DissIdent, a dissimilarity-based approach for identifying unknown UAVs in the airspace. This latter applied the first proposed contribution and was able to deal with critical challenges faced by the current ADD techniques, such as the detection and identification of unknown drones in the airspace.

From a discussion about how to represent the input signal features as dissimilarity vectors, we designed DissIdent as a multi-modal strategy. We conducted an experimental evaluation comparing DissIdent with supervised-based and clustering approaches. DissIdent overcame all the compared approaches in the detection task, addressing accuracy rates of about 99% on average for known signals and 94% for unknown signals. Regarding the identification of unknown signals, DissIdent also presented the highest rates, identifying correctly more than 93% of unauthorized drones, and also grouping them correctly.

The results also highlighted that the transformation from the feature to the dissimilarity space did not cause loss of information for the training/classification model as well for the deployment of the reference subsets, promoting an enhancement in the detection of known signals and a significant improvement regarding the unknown signals detection. Furthermore, the deployment and management of the reference subsets represented an important step in DissIdent’s performance, where keeping higher time intervals to spread the reference subsets leads to better results, as we observed configuring DissIdent with time intervals of one second and one minute.

Considering that ADD strategies are dedicated to UAV-based networks, the research questions of this dissertation are not applicable in this case. Nonetheless, DissIdent enhances the existing strategies, being empowered by the smart strategy using rhythmic-based features.

6.6.1 New Research Directions for ADD

In Section 6.3 we presented existing strategies to perform ADD, including different sources to extract features and computational methods to properly detect both known and unknown drones over the airspace. From that, we introduced the extraction of rhythm-based features from the drone’s propeller and the DissIdent proposal. However, other tasks are inherent to an enhanced ADD, representing open challenges in this field. In

a nutshell, signal pre-processing and DNN-related dissimilarity techniques from other research areas can be investigated for ADD, which are discussed as follows.

In signal pre-processing, signal denoising is a fundamental step before the feature extraction. For acoustic signals, specifically, some studies applied ML techniques to mitigate the noise effect, generating a clear sound to be processed by related classifiers [178]. Similarly, different methods can be applied to clear RF signals from environmental noise [179]. These strategies represent feasible approaches to be applied in an IoD environment, enhancing the pre-processing of incoming signals of ADD tasks.

Furthermore, our results highlighted that DissIdent can be considered a robust approach to the detection and identification of both known and unknown UAVs in the airspace. Nonetheless, other dissimilarity-based techniques have been designed and applied in other areas of knowledge, such as music classification and Natural Language Processing (NLP) [180, 181].

Deep Neural Networks (DNNs) represent a set of models with suitable performance in this regard [181]. For instance, Long Short-Term Memory (LSTM) networks can learn data dependencies used for anomaly detection to ensure security for a given system [182]. Siamese Neural Networks are projected to handle (dis)similarity properties of data [183]. In a nutshell, a Siamese Network is composed of two identical neural networks capable of learning the hidden representation of an input vector. They work parallelly and compare their outputs at the end. The generated output can be considered the semantic similarity between the projected representation of the two input vectors [183]. Likewise, transformer-based models are successfully applied in context-aware tasks (e.g. NLP) that demand fast processing and response, being more efficient than traditional DNNs, such as LSTMs [184]. Moreover, transformers do not require labeled data to train the referred models, meeting the characteristics of ADD for unknown data. All these techniques can be widely explored in future research to enhance ADD.

Chapter 7

Conclusion and Future Work

This chapter presents the final remarks of this dissertation as well as the future work to guide the investigation of further research. Therefore, Section 7.1 presents our concluding remarks, summarizing the contributions addressed with this dissertation. Last but not least, Section 7.2 presents future research directions based on the challenges highlighted in this dissertation.

7.1 Concluding Remarks

This dissertation studied the design of Protection Mechanisms for the Internet of Drones (IoD) paradigm, considering the particular characteristics of this environment. As an ignition point, we surveyed the current IoD research field regarding security and privacy aspects. We categorized the major groups of IoD-related attacks and protection mechanisms that can mitigate these threats. However, as IoD is a recent research field considering it as a robust mobile network, the protection mechanisms could potentially not meet the IoD characteristics and whether they can offer the same protection level or even be applied. This main challenge led us to the main research question of this dissertation: Can these existing protection mechanisms provide a proper level of security/privacy in IoD environments?

Aiming to guide this study, we propose a framework to guide the design of new protection mechanisms focusing on IoD. Overall, our contributions are grouped on three fronts: the design of Location Privacy Protection Mechanisms; the design of Anti-Jamming mechanisms; and the design of Automatic Drone Detection strategies. All the mechanisms are IoD-centered strategies such that they provide enhanced levels of security/privacy in this environment.

We designed three novel LPPMs. Two of them, namely t-MixDrones and MixRide, were based on the MZ concept, a well-known group of LPPMs in traditional mobile networks, such as VANETs. Through extensive performance evaluation, we demonstrated

that they overcame existing approaches in terms of location privacy considering dense IoD scenarios. Furthermore, MixRide can provide a better energy consumption compared to t-MixDrones since this mechanism promotes collaboration with ground vehicles, assigning real-time rides, saving the drones' energy while they can change their pseudonyms. The third proposed LPPM was TDG, based on a dummy-query strategy. This mechanism presented suitable levels of location privacy for sparse scenarios in IoD. Concluding this front, we also proposed IoDAPM, an RL-based approach for the dynamic assignment of LPPMs in IoD. IoDAPM emerged as a fundamental strategy to be deployed in IoD networks since the mechanism can apply the best mechanism given a set of environmental conditions, taking advantage of the best scenarios of each proposed LPPM, enhancing the provided QoS in this network.

In the second front, we designed the IoD-JAPM, an Anti-Jamming mechanism that considers the reformulation of drone path planning to avoid aerial regions compromised by the occurrence of a JA. We conducted an extensive performance evaluation comparing our proposal with existing strategies. This evaluation highlighted our assumption that IoD is a unique scenario. Summarily, the existing strategies can not handle properly the boundaries imposed by the airways inherent to IoD, failing to provide suitable security aspects to the drones. With IoD-JAPM, we could overcome these shortcomings.

In the third front, we enhance significantly the field of ADD strategies. The main challenge handled by our contributions was the lack of suitable strategies that can detect and identify unknown drones in the airspace just considering environmental sources, without a previous knowledge of these entities. Therefore, our main contribution was the DissIdent strategy, a distributed system that can detect and group unknown drones using ML and clustering techniques. DissIdent represents a novel category of ADD since they introduced the application of the dissimilarity concept in this field. Furthermore, we enhanced the performance of DissIdent through the investigation and application of rhythmic-based features processed from the sound generated by the drone's propeller. Through extensive performance evaluation, we demonstrated the robustness and validation of the proposed techniques.

In conclusion, all the proposed mechanisms and strategies presented a unanimous answer to the research questions raised in this dissertation: the existing protection mechanisms (considering the investigated research fronts) can not provide the same protection level of security/privacy to IoD environments when compared to traditional mobile networks. This answer justifies the need for the design of novel protection mechanisms in such a way they can be an adaptation of the existing ones, or even totally new strategies, designed from scratch.

7.2 Research Directions

The proposal of new protection mechanisms for IoD makes room for the observance of new challenges, properly described in the final remarks of each chapter. Therefore, these challenges point towards new research directions in the IoD security/privacy field. These directions are summarized as follows.

- *Study of the remained fronts of existing PMs*: although we explored three fronts in this dissertation, other mechanisms shall be investigated, as presented in Section 3.2. The conduction of new studies of Anti-Spoofing and Cryptographic-based mechanisms can lead to a better understanding regarding the impact of SA and HA over IoD and how to mitigate them properly. Specifically, the presence of airways represents an interesting aspect to exploit by these attacks, and therefore, how to counter the potential exploitation of this aspect.
- *Study of utility preservation for anonymized data*: in the information privacy context, data utility consists of upholding data semantics while applying privacy-preserving techniques. It involves finding a balance between protecting nodes' sensitive information and sustaining their meaningfulness for analysis or application [185]. The design of LPPMs was the major research front of this dissertation, therefore, data utility aspects shall be further investigated. Considering that the drone's trajectory and identity might be modified by the proposed LPPMs, evaluating how these modifications affect the information semantics by network authorities is a valuable research direction.
- *Study of QoS provisioning*: the design of the proposed mechanisms revealed trade-offs between the level of security/privacy and the provided QoS considering a given task. For instance, the application of MixRide, although can ensure location privacy, leads to flight delays due to the assigned ride. IoDAPM addressed suitable results to mitigate the trade-off effect, but this strategy is designed for LPPMs. Therefore, the design of smart strategies to optimize the inherent trade-off in a broader scope, considering the environmental conditions and different mechanisms, is a trending research area for a comprehensive investigation.
- *Enhancement of ADD through dissimilarity-related techniques*: as discussed in section 6.6.1, DissIdent is a first effort in applying the dissimilarity concept towards the detection and identification of unknown signals in IoD. However, related techniques from different research areas must be investigated in the ADD, such as LSTMs, Transformers, and Siamese Networks. Furthermore, there is room for the improvement of signal pre-processing tasks, mainly the ones related to noise reduction.

Considering the heterogeneous IoD environment, the quality of the incoming signal is a fundamental requirement to a proper detection and identification of the environmental elements.

- *Design of collaborative and smart protection mechanisms for IoD:* in this dissertation, we extensively discussed that each proposed mechanism addressed good results in specific conditions, countering different threats. Intuitively, deploying all these mechanisms together in a real-world IoD network will potentially overload the system, mainly because drones have SWaP limitations. Therefore, the application of protection mechanisms should occur based on the network needs. These challenges open a vast research field area, where the design of a symbiotic ecosystem of protection mechanisms for IoD whose activation may occur through the collaborative decision taken by the distributed network nodes.
- *Deployment and evaluation with testbed and real-world IoD infrastructures:* all the performance evaluations related to this dissertation were carried out through simulations, which is a proper strategy to obtain experimental insights about the performance of the proposed strategies. Nonetheless, the designed mechanisms shall migrate from the experimental field to the practical scope, through the experimentation with testbed and, therefore, in real-world IoD infrastructures. As a matter of fact, the lack of real-world infrastructure for IoD is a current issue, not only related to security/privacy concerns. Thus, it is necessary to establish partnerships between academia and industry to deploy robust IoD networks, embracing the airspace as an ITS and, hence, leveraging to the next generation of mobile networks.

References

- [1] Pietro Boccadoro, Domenico Striccoli, and Luigi Alfredo Grieco. An extensive survey on the internet of drones. *Ad Hoc Networks*, 122:102600, 2021.
- [2] Lailla MS Bine, Azzedine Boukerche, Linyer B Ruiz, and Antonio AF Loureiro. Leveraging urban computing with the internet of drones. *IEEE Internet of Things Magazine*, 5(1):160–165, 2022.
- [3] Matt Satell. 16 eye-opening drone stats [2023 update], Jul 2023. URL <https://www.phillybyair.com/blog/drone-stats/>.
- [4] Kai-Yun Tsao, Thomas Girdler, and Vassilios G Vassilakis. A survey of cyber security threats and solutions for uav communications and flying ad-hoc networks. *Ad Hoc Networks*, 133:102894, 2022.
- [5] Mirmojtaba Gharibi, Raouf Boutaba, and Steven L Waslander. Internet of drones. *IEEE Access*, 4:1148–1162, 2016.
- [6] Laith Abualigah, Ali Diabat, Putra Sumari, and Amir H Gandomi. Applications, deployments, and integration of internet of drones (iod): a review. *IEEE Sensors Journal*, 2021.
- [7] Chao Lin, Debiao He, Neeraj Kumar, Kim-Kwang Raymond Choo, Alexey Vinel, and Xinyi Huang. Security and privacy for the internet of drones: Challenges and solutions. *IEEE Communications Magazine*, 56(1):64–69, 2018.
- [8] Abdelouahid Derhab, Omar Cheikhrouhou, Azza Allouch, Anis Koubaa, Basit Qureshi, Mohamed Amine Ferrag, Leandros Maglaras, and Farrukh Aslam Khan. Internet of drones security: taxonomies, open issues, and future directions. *Vehicular Communications*, page 100552, 2022.
- [9] Yassine Mekdad, Ahmet Aris, Leonardo Babun, Abdeslam El Fergougui, Mauro Conti, Riccardo Lazzeretti, and A Selcuk Uluagac. A survey on security and privacy issues of uavs. *Computer Networks*, 224:109626, 2023.
- [10] Yueyan Zhi, Zhangjie Fu, Xingming Sun, and Jingnan Yu. Security and privacy issues of uav: A survey. *Mobile Networks and Applications*, pages 1–7, 2019.

- [11] Vincent Primault, Antoine Boutet, Sonia Ben Mokhtar, and Lionel Brunie. The long road to computational location privacy: A survey. *IEEE Communications Surveys & Tutorials*, 21(3):2772–2793, 2018.
- [12] Giovanni Geraci, Adrian Garcia-Rodriguez, M Mahdi Azari, Angel Lozano, Marco Mezzavilla, Symeon Chatzinotas, Yun Chen, Sundeep Rangan, and Marco Di Renzo. What will the future of uav cellular communications be? a flight from 5g to 6g. *IEEE communications surveys & tutorials*, 24(3):1304–1335, 2022.
- [13] Nahina Islam, Md Mamunur Rashid, Faezeh Pasandideh, Biplob Ray, Steven Moore, and Rajan Kadel. A review of applications and communication technologies for internet of things (iot) and unmanned aerial vehicle (uav) based sustainable smart farming. *Sustainability*, 13(4):1821, 2021.
- [14] Saeed Hamood Alsamhi, Alexey V Shvetsov, Santosh Kumar, Jahan Hassan, Mohammed A Alhartomi, Svetlana V Shvetsova, Radhya Sahal, and Ammar Hawbani. Computing in the sky: A survey on intelligent ubiquitous computing for uav-assisted 6g networks and industry 4.0/5.0. *Drones*, 6(7):177, 2022.
- [15] Qubeijian Wang, Hong-Ning Dai, Qiu Wang, Mahendra K Shukla, Wei Zhang, and Carlos Guedes Soares. On connectivity of uav-assisted data acquisition for underwater internet of things. *IEEE Internet of Things Journal*, 7(6):5371–5385, 2020.
- [16] Moayad Aloqaily, Ouns Bouachir, Azzedine Boukerche, and Ismaeel Al Ridhawi. Design guidelines for blockchain-assisted 5g-uav networks. *IEEE Network*, 35(1):64–71, 2021.
- [17] Sofiane Zaidi, Mohammed Atiquzzaman, and Carlos T Calafate. Internet of flying things (ioft): A survey. *Computer Communications*, 165:53–74, 2021.
- [18] Amira Chriki, Haifa Touati, Hichem Snoussi, and Farouk Kamoun. FANET: Communication, Mobility Models and Security Issues. *Computer Networks*, 163:106877, 2019. ISSN 1389-1286. doi: <https://doi.org/10.1016/j.comnet.2019.106877>. URL <https://www.sciencedirect.com/science/article/pii/S1389128618309034>.
- [19] Lailla MS Bine, Azzedine Boukerche, Linnyer B Ruiz, and Antonio AF Loureiro. A novel ant colony-inspired coverage path planning for internet of drones. *Computer Networks*, page 109963, 2023.
- [20] Juan Zhang, James F Campbell, Donald C Sweeney II, and Andrea C Hupman. Energy consumption models for delivery drones: A comparison and assessment. *Transportation Research Part D: Transport and Environment*, 90:102668, 2021.

-
- [21] John R Vacca. *Computer and Information Security Handbook*. Newnes, 2012.
- [22] Ekler P de Mattos, Augusto CSA Domingues, and Antonio AF Loureiro. Give me two points and i'll tell you who you are. In *2019 IEEE Intelligent Vehicles Symposium (IV)*, pages 1081–1087. IEEE, 2019.
- [23] Ekler P de Mattos, Augusto CSA Domingues, Bruno P Santos, Heitor S Ramos, and Antonio AF Loureiro. The impact of mobility on location privacy: A perspective on smart mobility. *IEEE Systems Journal*, 16(4):5509–5520, 2022.
- [24] Peng Wang, Bingliang Jiao, Lu Yang, Yifei Yang, Shizhou Zhang, Wei Wei, and Yanning Zhang. Vehicle re-identification in aerial imagery: Dataset and approach. In *Proceedings of the IEEE International Conference on Computer Vision*, pages 460–469, 2019.
- [25] Jitendra Bhatia, Ridham Dave, Heta Bhayani, Sudeep Tanwar, and Anand Nayyar. Sdn-based real-time urban traffic analysis in vanet environment. *Computer Communications*, 149:162–175, 2020.
- [26] Savio Sciancalepore, Omar Adel Ibrahim, Gabriele Oligeri, and Roberto Di Pietro. Pinch: An effective, efficient, and robust solution to drone detection via network traffic analysis. *Computer Networks*, 168:107044, 2020.
- [27] Haiquan Lu, Haiyang Zhang, Haibo Dai, Wei Wu, and Baoyun Wang. Proactive eavesdropping in uav-aided suspicious communication systems. *IEEE Transactions on Vehicular Technology*, 68(2):1993–1997, 2018.
- [28] Xiaoming Wang, Kai Li, Salil S Kanhere, Demin Li, Xiaolu Zhang, and Eduardo Tovar. Pele: Power efficient legitimate eavesdropping via jamming in uav communications. In *2017 13th International Wireless Communications and Mobile Computing Conference (IWCMC)*, pages 402–408. IEEE, 2017.
- [29] Liang Xiao, Caixia Xie, Minghui Min, and Weihua Zhuang. User-centric view of unmanned aerial vehicle transmission against smart attacks. *IEEE Transactions on Vehicular Technology*, 67(4):3420–3430, 2017.
- [30] Muhammad Asghar Khan, Insaf Ullah, Shibli Nisar, Fazal Noor, Ijaz Mansoor Qureshi, Fahim Ullah Khanzada, and Noor Ul Amin. An efficient and provably secure certificateless key-encapsulated signcryption scheme for flying ad-hoc network. *IEEE Access*, 8:36807–36828, 2020.
- [31] Chenxi Liu, Tony QS Quek, and Jemin Lee. Secure uav communication in the presence of active eavesdropper. In *2017 9th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, 2017.

- [32] Chenxi Liu, Jemin Lee, and Tony QS Quek. Safeguarding uav communications against full-duplex active eavesdropper. *IEEE Transactions on Wireless Communications*, 18(6):2919–2931, 2019.
- [33] Guangchi Zhang, Qingqing Wu, Miao Cui, and Rui Zhang. Securing uav communications via trajectory optimization. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.
- [34] Chaoqiong Fan, Huayi Liu, Bin Li, Chenglin Zhao, and Shiwen Mao. Adversarial game against hybrid attacks in uav communications with partial information. *IEEE Transactions on Vehicular Technology*, 2021.
- [35] Ruiqian Ma, Weiwei Yang, Yu Zhang, Jue Liu, and Hui Shi. Secure mmwave communication using uav-enabled relay and cooperative jammer. *IEEE Access*, 7: 119729–119741, 2019.
- [36] Tiep M Hoang, Nghia M Nguyen, and Trung Q Duong. Detection of eavesdropping attack in uav-aided wireless systems: Unsupervised learning with one-class svm and k-means clustering. *IEEE Wireless Communications Letters*, 2020.
- [37] Hongwu Liu, Sang-Jo Yoo, and Kyung Sup Kwak. Opportunistic relaying for low-altitude uav swarm secure communications with multiple eavesdroppers. *Journal of Communications and Networks*, 20(5):496–508, 2018.
- [38] Miao Cui, Guangchi Zhang, Qingqing Wu, and Derrick Wing Kwan Ng. Robust trajectory and transmit power design for secure uav communications. *IEEE Transactions on Vehicular Technology*, 67(9):9042–9046, 2018.
- [39] Honggu Kang, Jingon Joung, Jinhyun Ahn, and Joonhyuk Kang. Secrecy-aware altitude optimization for quasi-static uav base station without eavesdropper location information. *IEEE Communications Letters*, 23(5):851–854, 2019.
- [40] Hongjiang Lei, Di Wang, Ki-Hong Park, Imran Shafique Ansari, Jing Jiang, Gaofeng Pan, and Mohamed-Slim Alouini. Safeguarding uav iot communication systems against randomly located eavesdroppers. *IEEE Internet of Things Journal*, 2020.
- [41] Zhichao Sheng, Hoang D Tuan, AA Nasir, Trung Q Duong, and H Vincent Poor. Secure uav-enabled communication using han-kobayashi signaling. *IEEE Transactions on Wireless Communications*, 2020.
- [42] Huici Wu, Yang Wen, Jiazhen Zhang, Zhiqing Wei, Ning Zhang, and Xiaofeng Tao. Energy-efficient and secure air-to-ground communication with jittering uav. *IEEE Transactions on Vehicular Technology*, 2020.

- [43] Shehzad Ashraf Chaudhry, Khalid Yahya, Marimuthu Karuppiah, Rupak Kharel, Ali Kashif Bashir, and Yousaf bin Zikria. Gcacs-iod: A certificate based generic access control scheme for internet of drones. *Computer Networks*, page 107999, 2021.
- [44] Yufang Gao, Yang Wu, Zhichao Cui, Wendong Yang, and Ning Li. Anti-jamming trajectory and power design for cognitive uav communications. In *2021 International Wireless Communications and Mobile Computing (IWCMC)*, pages 1370–1375. IEEE, 2021.
- [45] Yifan Xu, Guochun Ren, Jin Chen, Yunpeng Luo, Luliang Jia, Xin Liu, Yang Yang, and Yuhua Xu. A one-leader multi-follower bayesian-stackelberg game for anti-jamming transmission in uav communication networks. *IEEE Access*, 6:21697–21709, 2018.
- [46] Zhiwei Li, Yu Lu, Yun Shi, Zengguang Wang, Wenxin Qiao, and Yicen Liu. A dyna-q-based solution for uav networks against smart jamming attacks. *Symmetry*, 11(5):617, 2019.
- [47] Bertold Van den Bergh and Sofie Pollin. Keeping uavs under control during gps jamming. *IEEE Systems Journal*, 13(2):2010–2021, 2019.
- [48] Maanak Gupta, Mahmoud Abdelsalam, Sajad Khorsandroo, and Sudip Mittal. Security and privacy in smart farming: Challenges and opportunities. *IEEE Access*, 8:34564–34584, 2020.
- [49] Maha Sliti, Walid Abdallah, and Nouredine Boudriga. Jamming attack detection in optical uav networks. In *2018 20th International Conference on Transparent Optical Networks (ICTON)*, pages 1–5. IEEE, 2018.
- [50] Zihan Lin, Xiaozhen Lu, Canhuang Dai, Geyi Sheng, and Liang Xiao. Reinforcement learning based uav trajectory and power control against jamming. In *International Conference on Machine Learning for Cyber Security*, pages 336–347. Springer, 2019.
- [51] Yang Wu, Weiwei Yang, Xinrong Guan, and Qingqing Wu. Uav-enabled relay communication under malicious jamming: Joint trajectory and transmit power optimization. *IEEE Transactions on Vehicular Technology*, 2021.
- [52] Ali Krayani, Atm Shafiu Alam, Lucio Marcenaro, Arumugam Nallanathan, and Carlo Regazzoni. Automatic jamming signal classification in cognitive uav radios. *IEEE Transactions on Vehicular Technology*, 71(12):12972–12988, 2022.
- [53] Daniel P Shepard, Jahshan A Bhatti, Todd E Humphreys, and Aaron A Fansler. Evaluation of smart grid and civilian uav vulnerability to gps spoofing attacks. In *Radionavigation Laboratory Conference Proceedings*, 2012.

- [54] Tao Zhang and Quanyan Zhu. Strategic defense against deceptive civilian gps spoofing of unmanned aerial vehicles. In *International Conference on Decision and Game Theory for Security*, pages 213–233. Springer, 2017.
- [55] Kai Jansen, Matthias Schäfer, Daniel Moser, Vincent Lenders, Christina Pöpper, and Jens Schmitt. Crowd-gps-sec: Leveraging crowdsourcing to detect and localize gps spoofing attacks. In *2018 IEEE Symposium on Security and Privacy (SP)*, pages 1018–1031. IEEE, 2018.
- [56] Ke-Wen Huang and Hui-Ming Wang. Combating the control signal spoofing attack in uav systems. *IEEE Transactions on Vehicular Technology*, 67(8):7769–7773, 2018.
- [57] Yan Guo, Meiping Wu, Kanghua Tang, Junbo Tie, and Xian Li. Covert spoofing algorithm of uav based on gps/ins-integrated navigation. *IEEE Transactions on Vehicular Technology*, 68(7):6557–6564, 2019.
- [58] Mingzhu Zhang, Yu Chen, Xiaofeng Tao, and Izzat Darwazeh. Power allocation for proactive eavesdropping with spoofing relay in uav systems. In *2019 26th International Conference on Telecommunications (ICT)*, pages 102–107. IEEE, 2019.
- [59] Arslan Shafique, Abid Mehmood, and Mourad Elhadef. Detecting signal spoofing attack in uavs using machine learning models. *IEEE Access*, 9:93803–93815, 2021.
- [60] Abdel Rahman Eldosouky, Aidin Ferdowsi, and Walid Saad. Drones in distress: A game-theoretic countermeasure for protecting uavs against gps spoofing. *IEEE Internet of Things Journal*, 2020.
- [61] Drew Davidson, Hao Wu, Rob Jellinek, Vikas Singh, and Thomas Ristenpart. Controlling uavs with sensor input spoofing attacks. In *10th Workshop on Offensive Technologies*, pages 1–11, Austin, TX, 2016. USENIX Association. URL <https://www.usenix.org/conference/woot16/workshop-program/presentation/davidson>.
- [62] Daniel Mendes, Naghmeh Ivaki, and Henrique Madeira. Effects of gps spoofing on unmanned aerial vehicles. In *2018 IEEE 23rd Pacific Rim International Symposium on Dependable Computing (PRDC)*, pages 155–160. IEEE, 2018.
- [63] Zhiwei Feng, Nan Guan, Mingsong Lv, Weichen Liu, Qingxu Deng, Xue Liu, and Wang Yi. An efficient uav hijacking detection method using onboard inertial measurement unit. *ACM Transactions on Embedded Computing Systems (TECS)*, 17(6):1–19, 2019.
- [64] Juhwan Noh, Yujin Kwon, Yunmok Son, Hocheol Shin, Dohyun Kim, Jaeyeong Choi, and Yongdae Kim. Tractor beam: Safe-hijacking of consumer drones with

- adaptive gps spoofing. *ACM Transactions on Privacy and Security (TOPS)*, 22(2): 1–26, 2019.
- [65] Jason McNeely, Michael Hatfield, Abir Hasan, and Nusrat Jahan. Detection of uav hijacking and malfunctions via variations in flight data statistics. In *2016 IEEE International Carnahan Conference on Security Technology (ICCST)*, pages 1–8. IEEE, 2016.
- [66] Dan Mototolea and Comelis Stolk. Software defined radio for analyzing drone communication protocols. In *2018 International Conference on Communications (COMM)*, pages 485–490. IEEE, 2018.
- [67] Jorg Daubert, Dhanasekar Boopalan, Max Mühlhäuser, and Emmanouil Vasilo-manolakis. Honeydrone: A medium-interaction unmanned aerial vehicle honeypot. In *NOMS 2018-2018 IEEE/IFIP Network Operations and Management Symposium*, pages 1–6, Taipei, Taiwan, 2018. IEEE, IEEE.
- [68] Johann-Sebastian Pleban, Ricardo Band, and Reiner Creutzburg. Hacking and securing the ar. drone 2.0 quadcopter: Investigations for improving the security of a toy. In *Mobile Devices and Multimedia: Enabling Technologies, Algorithms, and Applications 2014*, volume 9030, page 90300L. International Society for Optics and Photonics, 2014.
- [69] Jinyeong Kang and Inwhae Joe. Security vulnerability analysis of wi-fi connection hijacking on the linux-based robot operating system for drone systems. In *International Conference on Parallel and Distributed Computing: Applications and Technologies*, pages 473–482. Springer, 2018.
- [70] Garrett Jares and John Valasek. Investigating malware-in-the-loop autopilot attack using falsification of sensor data. In *2021 International Conference on Unmanned Aircraft Systems (ICUAS)*, pages 1268–1276. IEEE, 2021.
- [71] Junia Valente and Alvaro A Cardenas. Understanding security threats in consumer drones through the lens of the discovery quadcopter family. In *Proceedings of the 2017 Workshop on Internet of Things Security and Privacy*, pages 31–36, 2017.
- [72] Vasily Desnitsky and Igor Kotenko. Simulation and assessment of battery depletion attacks on unmanned aerial vehicles for crisis management infrastructures. *Simulation Modelling Practice and Theory*, 107:102244, 2021.
- [73] Hamssa Hasrouny, Abed Ellatif Samhat, Carole Bassil, and Anis Laouiti. Vanet security challenges and solutions: A survey. *Vehicular Communications*, 7:7–20, 2017.

- [74] Mohamed Amine Ferrag, Leandros Maglaras, Antonios Argyriou, Dimitrios Kosmanos, and Helge Janicke. Security for 4g and 5g cellular networks: A survey of existing authentication and privacy-preserving schemes. *Journal of Network and Computer Applications*, 101:55–82, 2018.
- [75] Rei Yamazaki, Masashi Yoshida, and Hiroshi Shigeno. A dynamic mix-zone scheme considering communication delay for location privacy in vehicular networks. In *2021 IEEE International Conference on Pervasive Computing and Communications Workshops and other Affiliated Events (PerCom Workshops)*, pages 245–250, 2021.
- [76] Youhuizi Li, Yuyu Yin, Xu Chen, Jian Wan, Gangyong Jia, and Kewei Sha. A secure dynamic mix zone pseudonym changing scheme based on traffic context prediction. *IEEE Transactions on Intelligent Transportation Systems*, 2021.
- [77] Lei Hou, Nianmin Yao, Zhimao Lu, Furui Zhan, and Zhimin Liu. Tracking based mix-zone location privacy evaluation in vanet. *IEEE Transactions on Vehicular Technology*, 70(10):10957–10969, 2021.
- [78] Zhixiang Zhang, Tianyi Feng, Wai-Choong Wong, and Biplab Sikdar. A geo-indistinguishable context-based mix strategy for trajectory protection in vanets. *IEEE Transactions on Vehicular Technology*, 2023.
- [79] Abdueli Paulo Mdee, Muhammad Toaha Raza Khan, Junho Seo, and Dongkyun Kim. Security compliant and cooperative pseudonyms swapping for location privacy preservation in vanets. *IEEE Transactions on Vehicular Technology*, 2023.
- [80] Alisson Renan Svaigen, Heitor S Ramos, Linnyer B Ruiz, and Antonio AF Loureiro. Dynamic temporal mix-zone placement approach for location-based services privacy. In *2019 IEEE Latin-American Conference on Communications (LATINCOM)*, pages 1–6, 2019.
- [81] Ashwin Machanavajjhala, Johannes Gehrke, Daniel Kifer, and Muthuramakrishnan Venkitasubramaniam. L-diversity: Privacy beyond k-anonymity. In *22nd International Conference on Data Engineering (ICDE'06)*, pages 24–24. IEEE, 2006.
- [82] Muhammad Arif, Guojun Wang, and Tao Peng. Track me if you can? query based dual location privacy in vanets for v2v and v2i. In *IEEE TrustCom/BigDataSE*, pages 1091–1096. IEEE, 2018.
- [83] Jian Kang, Doug Steiert, Dan Lin, and Yanjie Fu. Movewithme: Location privacy preservation for smartphone users. *Transactions on Information Forensics and Security*, 15:711–724, 2019.

- [84] Ping Zhao, Wuwu Liu, Guanglin Zhang, Zongpeng Li, and Lin Wang. Preserving privacy in wifi localization with plausible dummy locations. *IEEE Transactions on VT*, 69(10):11909–11925, 2020.
- [85] Zongda Wu, Guiling Li, Shigen Shen, Xinze Lian, Enhong Chen, and Guandong Xu. Constructing dummy query sequences to protect location privacy and query privacy in location-based services. *WWW*, 24(1):25–49, 2021.
- [86] Zhibo Wang, Yuting Huang, Xinkai Wang, Ju Ren, Qian Wang, and Libing Wu. Socialrecruiter: Dynamic incentive mechanism for mobile crowdsourcing worker recruitment with social networks. *IEEE Transactions on Mobile Computing*, 20(5):2055–2066, 2020.
- [87] Hossein Pirayesh and Huacheng Zeng. Jamming attacks and anti-jamming strategies in wireless networks: A comprehensive survey. *IEEE communications surveys & tutorials*, 24(2):767–809, 2022.
- [88] Nishat I Mowla, Nguyen H Tran, Inshil Doh, and Kijoon Chae. Afrl: Adaptive federated reinforcement learning for intelligent jamming defense in fanet. *Journal of Communications and Networks*, 22(3):244–258, 2020.
- [89] Liang Xiao, Xiaozhen Lu, Dongjin Xu, Yuliang Tang, Lei Wang, and Weihua Zhuang. Uav relay in vanets against smart jamming with reinforcement learning. *IEEE Trans. on VT*, 67(5):4087–4097, 2018.
- [90] Hichem Sedjelmaci and Sidi Mohammed Senouci. A hierarchical detection and response system to enhance security against lethal cyber-attacks in uav networks. *IEEE Trans. on SM&C*, 48(9):1594–1606, 2017.
- [91] FAA. Uas remote identification, 2020. URL https://www.faa.gov/uas/getting-started/remote_id. Access date: June 27, 2022.
- [92] Harini Kolamunna, Thilini Dahanayaka, Junye Li, Suranga Seneviratne, Kanchana Thilakaratne, Albert Y Zomaya, and Aruna Seneviratne. Droneprint: Acoustic signatures for open-set drone detection and identification with online data. *Proceedings of the ACM on Interactive, Mobile, Wearable and Ubiquitous Technologies*, 5(1):1–31, 2021.
- [93] Ibrahim Nemer, Tarek Sheltami, Irfan Ahmad, Ansar Ul-Haque Yasar, and Mohammad AR Abdeen. Rf-based uav detection and identification using hierarchical learning approach. *Sensors*, 21(6):1947, 2021.
- [94] Ghulam E Mustafa Abro, Saiful Azrin BM Zulkifli, Rana Javed Masood, Vijanth Sagayan Asirvadam, and Anis Laouti. Comprehensive review of uav detection,

- security, and communication advancements to prevent threats. *Drones*, 6(10):284, 2022.
- [95] Angelo Coluccia, Alessio Fascista, Arne Schumann, Lars Sommer, Anastasios Dimou, Dimitrios Zarpalas, Fatih Cagatay Akyon, Ogulcan Eryuksel, Kamil Anil Ozfuttu, Sinan Onur Altinuc, et al. Drone-vs-bird detection challenge at iee avss2021. In *IEEE AVSS*, pages 1–8. IEEE, 2021.
- [96] Rui Sun, Wenyu Zhang, Jiazhu Zheng, and Washington Yotto Ochieng. Gnss/ins integration with integrity monitoring for uav no-fly zone management. *Remote Sensing*, 12(3), 2020.
- [97] Joe Khalife, Mahdi Maaref, and Zaher M Kassas. Opportunistic autonomous integrity monitoring for enhanced uav safety. *IEEE Aerospace and Electronic Systems Magazine*, 2022.
- [98] Mahdi Maaref and Zaher M Kassas. Autonomous integrity monitoring for vehicular navigation with cellular signals of opportunity and an imu. *IEEE Trans. on Intelligent Transportation Systems*, 2021.
- [99] Ghilas Aissou, Hadjar Ould Slimane, Selma Benouadah, and Naima Kaabouch. Tree-based supervised machine learning models for detecting gps spoofing attacks on uas. In *2021 IEEE UEMCON*, pages 649–653, 2021.
- [100] Tala Talaei Khoei, Shereen Ismail, and Naima Kaabouch. Dynamic selection techniques for detecting gps spoofing attacks on uavs. *Sensors*, 22(2), 2022.
- [101] Prosanta Gope, Owen Millwood, and Neetesh Saxena. A provably secure authentication scheme for rfid-enabled uav applications. *Computer Communications*, 166: 19–25, 2021.
- [102] Yuyang Cheng, Shiyuan Xu, Miao Zang, and Weimin Kong. Lppa: a lightweight privacy-preserving authentication scheme for the internet of drones. In *2021 IEEE 21st International Conference on Communication Technology (ICCT)*, pages 656–661. IEEE, 2021.
- [103] Tejasvi Alladi, Vinay Chamola, Nishad Sahu, and Mohsen Guizani. Applications of blockchain in unmanned aerial vehicles: A review. *Vehicular Communications*, 23, 2020.
- [104] Tejasvi Alladi, Gaurang Bansal, Vinay Chamola, Mohsen Guizani, et al. Secau-thuav: A novel authentication scheme for uav-ground station and uav-uav communication. *IEEE Transactions on VT*, 69(12):15068–15077, 2020.

- [105] Abdelouahid Derhab, Omar Cheikhrouhou, Azza Allouch, Anis Koubaa, Basit Qureshi, Mohamed Amine Ferrag, Leandros Maglaras, and Farrukh Aslam Khan. Internet of drones security: Taxonomies, open issues, and future directions. *Vehicular Communications*, 39:100552, 2023. doi: <https://doi.org/10.1016/j.vehcom.2022.100552>.
- [106] Alastair R Beresford and Frank Stajano. Mix zones: User privacy in location-aware services. In *IEEE Annual Conference on Pervasive Computing and Communications Workshops, 2004. Proceedings of the Second*, pages 127–131. IEEE, 2004.
- [107] Qasim Ali Arain, Imran Memon, Zhongliang Deng, Muhammad Hammad Memon, Farman Ali Mangi, and Asma Zubedi. Location monitoring approach: multiple mix-zones with location privacy protection based on traffic flow over road networks. *Multimedia Tools and Applications*, 77(5):5563–5607, 2018.
- [108] Balaji Palanisamy and Ling Liu. Mobimix: Protecting location privacy with mix-zones over road networks. In *2011 IEEE 27th International Conference on Data Engineering*, pages 494–505, 2011.
- [109] Balaji Palanisamy and Ling Liu. Attack-resilient mix-zones over road networks: Architecture and algorithms. *IEEE Transactions on Mobile Computing*, 14(3):495–508, 2014.
- [110] Julien Freudiger, Reza Shokri, and Jean-Pierre Hubaux. On the optimal placement of mix zones. In *International Symposium on Privacy Enhancing Technologies Symposium*, pages 216–234, 2009.
- [111] Xinxin Liu, Han Zhao, Miao Pan, Hao Yue, Xiaolin Li, and Yuguang Fang. Traffic-aware multiple mix zone placement for protecting location privacy. In *2012 Proceedings IEEE INFOCOM*, pages 972–980, 2012.
- [112] Imran Memon, Qasim Ali, Asma Zubedi, and Farman Ali Mangi. Dpmm: Dynamic pseudonym-based multiple mix-zones generation for mobile traveler. *Multimedia Tools and Applications*, 76(22):24359–24388, 2017.
- [113] Imran Memon. Distance and clustering-based energy-efficient pseudonyms changing strategy over road network. *International Journal of Communication Systems*, 31(11):e3704, 2018.
- [114] Imran Memon, Hamid Turab Mirza, Qasim Ali Arain, and Hina Memon. Multiple mix zones de-correlation trajectory privacy model for road network. *Telecommunication Systems*, 70(4):557–582, 2019.

-
- [115] Min Yang, Yong Feng, Xiaodong Fu, and Qian Qian. Location privacy preserving scheme based on dynamic pseudonym swap zone for internet of vehicles. *International Journal of Distributed Sensor Networks*, 15(7), 2019.
- [116] Nirupama Ravi, C Mani Krishna, and Israel Koren. Enhancing vehicular anonymity in its: A new scheme for mix zones and their placement. *IEEE Transactions on Vehicular Technology*, 68(11):10372–10381, 2019.
- [117] Mohammad Khodaei. Cooperative location privacy in vehicular networks: Why simple mix zones are not enough. *IEEE Internet of Things Journal*, 8(10):7985–8004, 2020.
- [118] Xinghua Li, Huijuan Zhang, Yanbing Ren, Siqi Ma, Bin Luo, Jian Weng, Jianfeng Ma, and Xiaoming Huang. Papu: Pseudonym swap with provable unlinkability based on differential privacy in vanets. *IEEE Internet of Things Journal*, 7(12):11789–11802, 2020.
- [119] Imran Memon, Hina Memon, and Qasim Ali Arain. Pseudonym changing strategy with mix zones based authentication protocol for location privacy in road networks. *Wireless Personal Communications*, 116(4):3309–3329, 2020.
- [120] Chengzhe Lai, Qian Li, Haibo Zhou, and Dong Zheng. Srsp: a secure and reliable smart parking scheme with dual privacy preservation. *IEEE Internet of Things Journal*, 2020.
- [121] Jawad Naveed Yasin, Sherif Abdelmonem Sayed Mohamed, Mohammad-Hashem Haghbayan, Jukka Heikkonen, Hannu Tenhunen, Muhammad Mehboob Yasin, and Juha Plosila. Energy-efficient formation morphing for collision avoidance in a swarm of drones. *IEEE Access*, 8:170681–170695, 2020.
- [122] Gamil Ahmed, Tarek Sheltami, Mohamed Deriche, and Ansar Yasar. An energy efficient iod static and dynamic collision avoidance approach based on gradient optimization. *Ad Hoc Networks*, 118:102519, 2021.
- [123] Adarsh Kumar, Rajalakshmi Krishnamurthi, Anand Nayyar, Ashish Kr Luhach, Mohammad S Khan, and Anuraj Singh. A novel software-defined drone network (sddn)-based collision avoidance strategies for on-road traffic monitoring and management. *Vehicular Communications*, 28:100313, 2021.
- [124] Arpan Kumar Kar. Bio inspired computing—a review of algorithms and scope of applications. *Expert Systems with Applications*, 59:20–32, 2016.
- [125] Marco Dorigo, Mauro Birattari, and Thomas Stutzle. Ant colony optimization. *IEEE computational intelligence magazine*, 1(4):28–39, 2006.

- [126] Jingyun Feng, Zhi Liu, Celimuge Wu, and Yusheng Ji. Ave: Autonomous vehicular edge computing framework with aco-based scheduling. *IEEE Transactions on Vehicular Technology*, 66(12):10660–10675, 2017.
- [127] Khac-Hoai Nam Bui and Jason J Jung. Aco-based dynamic decision making for connected vehicles in iot system. *IEEE Transactions on Industrial Informatics*, 15(10):5648–5655, 2019.
- [128] Angelo Trotta, Fabio D Andreagiovanni, Marco Di Felice, Enrico Natalizio, and Kaushik Roy Chowdhury. When uavs ride a bus: Towards energy-efficient city-scale video surveillance. In *IEEE INFOCOM*, pages 1043–1051, 2018.
- [129] Hailong Huang, Andrey V Savkin, and Chao Huang. A new parcel delivery system with drones and a public train. *Journal of Intelligent & Robotic Systems*, 100(3):1341–1354, 2020.
- [130] Yan Pan, Shining Li, Qianwu Chen, Nan Zhang, Tao Cheng, Zhigang Li, Bin Guo, Qingye Han, and Ting Zhu. Efficient schedule of energy-constrained uav using crowdsourced buses in last-mile parcel delivery. *ACM IMWUT*, 5(1):1–23, 2021.
- [131] Ziyi Lu, Na Yu, and Xuehe Wang. Incentive mechanism and path planning for unmanned aerial vehicle (uav) hitching over traffic networks. *Future Generation Computer Systems*, 2023.
- [132] Hailong Huang and Andrey V Savkin. Aerial surveillance in cities: When uavs take public transportation vehicles. *IEEE Transactions on Automation Science and Engineering*, 2022.
- [133] Nouredine Lasla, Hakim Ghazzai, Hamid Menouar, and Yehia Massoud. Exploiting land transport to improve the uav’s performances for longer mission coverage in smart cities. In *2019 IEEE 89th Vehicular Technology Conference (VTC2019-Spring)*, pages 1–7. IEEE, 2019.
- [134] Daniel Dias and Luís Henrique Maciel Kosmowski Costa. CRAWDAD dataset coppe-ufrrj/riobuses (v. 2018-03-19). Downloaded from <https://crawdad.org/coppe-ufrrj/RioBuses/20180319>, March 2018.
- [135] Juan Zhang, James F Campbell, Donald C Sweeney II, and Andrea C Hupman. Energy consumption models for delivery drones: A comparison and assessment. *Transport and Environment*, 90, 2021.
- [136] Walton Pereira Coutinho, Maria Battarra, and Jörg Fliege. The unmanned aerial vehicle routing and trajectory optimisation problem, a taxonomic review. *Computers & Industrial Engineering*, 120:116–128, 2018.

- [137] B. Niu, Q. Li, X. Zhu, and G. Cao. Enhancing privacy through caching in location-based services. In *INFOCOM*, pages 1017–1025. IEEE, 2015.
- [138] Hai Liu, Xinghua Li, Hui Li, Jianfeng Ma, and Xindi Ma. Spatiotemporal correlation-aware dummy-based privacy protection scheme for location-based services. In *IEEE INFOCOM*, pages 1–9. IEEE, 2017.
- [139] Vincent Bindschaedler and Reza Shokri. Synthesizing plausible privacy-preserving location traces. In *IEEE SP*, pages 546–563. IEEE, 2016.
- [140] Tun-Hao You, Wen-Chih Peng, and Wang-Chien Lee. Protecting moving trajectories with dummies. In *IEEE MDM*, pages 278–282. IEEE, 2007.
- [141] Roniel S De Sousa, Azzedine Boukerche, and Antonio AF Loureiro. Vehicle trajectory similarity: Models, methods, and applications. *ACM CSUR*, 53(5):1–32, 2020.
- [142] Nguyen Cong Luong, Dinh Thai Hoang, Shimin Gong, Dusit Niyato, Ping Wang, Ying-Chang Liang, and Dong In Kim. Applications of deep reinforcement learning in communications and networking: A survey. *IEEE Communications Surveys & Tutorials*, 21(4):3133–3174, 2019.
- [143] M Mehdi Afsar, Trafford Crump, and Behrouz Far. Reinforcement learning based recommender systems: A survey. *ACM Computing Surveys*, 55(7):1–38, 2022.
- [144] Ammar Haydari and Yasin Yilmaz. Deep reinforcement learning for intelligent transportation systems: A survey. *IEEE Transactions on Intelligent Transportation Systems*, 23(1):11–32, 2020.
- [145] FAA and DOT. Remote identification of unmanned aircraft, 2021. URL https://www.faa.gov/sites/faa.gov/files/2021-08/RemoteID_Final_Rule.pdf. 1–470. RIN 2120–AL31. Access date: June 27, 2022.
- [146] Pietro Tedeschi, Savio Sciancalepore, and Roberto Di Pietro. Arid: Anonymous remote identification of unmanned aerial vehicles. In *Annual Computer Security Applications Conference*, pages 207–218, 2021.
- [147] Omkar Mujumdar, Haluk Celebi, Ismail Guvenc, Mihail Sichitiu, Sunghyun Hwang, and Kyu-Min Kang. Use of lora for uav remote id with multi-user interference and different spreading factors. In *VTC2021-Spring*, pages 1–7. IEEE, 2021.
- [148] Vanessa Kuroda, Maxim Egorov, Steven Munn, and Antony Evans. Unlicensed technology assessment for uas communications. In *ICNS*. IEEE, 2020.

- [149] Ethan Murrell, Zach Walker, Eric King, and Kamesh Namuduri. Remote id and vehicle-to-vehicle communications for unmanned aircraft system traffic management. In *International Workshop on Communication Technologies for Vehicles*, pages 194–202. Springer, 2020.
- [150] Bin Duo, Qingqing Wu, Xiaojun Yuan, and Rui Zhang. Anti-jamming 3d trajectory design for uav-enabled wireless sensor networks under probabilistic los channel. *IEEE Trans. on VT*, 69(12):16288–16293, 2020.
- [151] Haichao Wang, Jin Chen, Guoru Ding, and Jiachen Sun. Trajectory planning in uav communication with jamming. In *2018 10th International Conference on Wireless Communications and Signal Processing (WCSP)*, pages 1–6. IEEE, 2018.
- [152] Yang Wu, Wenlu Fan, Weiwei Yang, Xiaoli Sun, and Xinrong Guan. Robust trajectory and communication design for multi-uav enabled wireless networks in the presence of jammers. *IEEE Access*, 8:2893–2905, 2019.
- [153] Xiaozhen Lu, Dongjin Xu, Liang Xiao, Lei Wang, and Weihua Zhuang. Anti-jamming communication game for uav-aided vanets. In *GLOBECOM 2017-2017 IEEE Global Communications Conference*, pages 1–6. IEEE, 2017.
- [154] Lailla M. S. Bine, Azzedine Boukerche, Linnyer B. Ruiz, and Antonio A. F. Loureiro. Iodscf: A store-carry-forward routing protocol for joint bus networks and internet of drones. In *2022 IEEE 42nd International Conference on Distributed Computing Systems (ICDCS)*, pages 950–960, 2022. doi: 10.1109/ICDCS54860.2022.00096.
- [155] Quan Quan, Rao Fu, Mengxin Li, Donghui Wei, Yan Gao, and Kai-Yuan Cai. Practical distributed control for vtol uavs to pass a virtual tube. *IEEE Trans. on Intelligent Vehicles*, 7(2):342–353, 2022. doi: 10.1109/TIV.2021.3123110.
- [156] Minsu Kim, Seongjun Kim, and Jemin Lee. Securing communications with friendly unmanned aerial vehicle jammers. *IEEE Transactions on Vehicular Technology*, 70(2):1972–1977, 2021.
- [157] Alisson R Svaigen, Azzedine Boukerche, Linnyer B Ruiz, and Antonio AF Loureiro. Um mecanismo de proteção ciente de vias aéreas contra jamming attacks para a internet dos drones. In *Anais do XL Simpósio Brasileiro de Redes de Computadores e Sistemas Distribuídos*, pages 405–418. SBC, 2022.
- [158] Qingqing Wu, Weidong Mei, and Rui Zhang. Safeguarding wireless network with uavs: A physical layer security perspective. *IEEE Wireless Communications*, 26(5): 12–18, 2019.

- [159] Nishat I Mowla, Nguyen H Tran, Inshil Doh, and Kijoon Chae. Federated learning-based cognitive detection of jamming attack in flying ad-hoc network. *IEEE Access*, 8:4338–4350, 2019.
- [160] Ali Krayani, Mohamad Baydoun, Lucio Marcenaro, Yue Gao, and Carlo S Regazzoni. Smart jammer detection for self-aware cognitive uav radios. In *31st ISPIIMRC*, pages 1–7. IEEE, 2020.
- [161] Mohammad F Al-Sa’d, Abdulla Al-Ali, Amr Mohamed, Tamer Khattab, and Aiman Erbad. Rf-based drone detection and identification using deep learning approaches: An initiative towards a large open source drone database. *Future Generation Computer Systems*, 100:86–97, 2019.
- [162] Rafael B Mangolin et al. A multimodal approach for multi-label movie genre classification. *Multimedia Tools and Applications*, pages 1–26, 2020.
- [163] Absalom E Ezugwu, Abiodun M Ikotun, Olaide O Oyelade, Laith Abualigah, Jeffery O Agushaka, Christopher I Eke, and Andronicus A Akinyelu. A comprehensive survey of clustering algorithms: State-of-the-art machine learning applications, taxonomy, challenges, and future research prospects. *Engineering Applications of Artificial Intelligence*, 110:104743, 2022.
- [164] Yandre MG Costa, Diego Bertolini, Alceu S Britto, George DC Cavalcanti, and Luiz ES Oliveira. The dissimilarity approach: a review. *Artificial Intelligence Review*, 53(4):2783–2808, 2020.
- [165] Allistair Moses, Matthew J Rutherford, and Kimon P Valavanis. Radar-based detection and identification for miniature air vehicles. In *2011 IEEE International Conference on Control Applications (CCA)*, pages 933–940. IEEE, 2011.
- [166] Dmitrii Solomitckii, Margarita Gapeyenko, Vasili Semkin, Sergey Andreev, and Yevgeni Koucheryavy. Technologies for efficient amateur drone detection in 5g millimeter-wave cellular infrastructure. *IEEE Communications Magazine*, 56(1): 43–50, 2018.
- [167] Chaoqun Yang, Zexian Wu, Xianyu Chang, Xiufang Shi, Junfeng Wo, and Zhiguo Shi. Doa estimation using amateur drones harmonic acoustic signals. In *2018 IEEE 10th Sensor Array and Multichannel Signal Processing Workshop (SAM)*, pages 587–591. IEEE, 2018.
- [168] Sayan Mandal, Lei Chen, Vishwa Alaparthi, and Mary L Cummings. Acoustic detection of drones through real-time audio attribute prediction. In *AIAA Scitech 2020 Forum*, page 0491, 2020.

- [169] Sara Al-Emadi, Abdulla Al-Ali, Amr Mohammad, and Abdulaziz Al-Ali. Audio based drone detection and identification using deep learning. In *2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, pages 459–464. IEEE, 2019.
- [170] Eren Unlu, Emmanuel Zenou, and Nicolas Rivière. Using shape descriptors for uav detection. *Electronic Imaging*, 2018(9):128–1, 2018.
- [171] Fredrik Svanström, Fernando Alonso-Fernandez, and Cristofer Englund. Drone detection and tracking in real-time by fusion of different sensing modalities. *Drones*, 6(11):317, 2022.
- [172] Loris Nanni, Yandre MG Costa, Diego R Lucio, Carlos N Silla, and Sheryl Brahnam. Combining visual and acoustic features for bird species classification. In *2016 IEEE 28th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 396–401. IEEE, 2016.
- [173] Marco Aurelio Deoldoto Paulino, Alceu Souza Britto Junior, Alisson Renan Svaigen, Linnyer Beatrys Ruiz Aylon, Luiz Eduardo Soares de Oliveira, et al. A Brazilian Speech Database. In *2018 IEEE 30th International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 234–241. IEEE, 2018.
- [174] Alisson R Svaigen, Lailla MS Bine, Gisele L Pappa, Linnyer B Ruiz, and Antonio AF Loureiro. Automatic drone identification through rhythm-based features for the internet of drones. In *2021 IEEE 33rd International Conference on Tools with Artificial Intelligence (ICTAI)*, pages 1417–1421. IEEE, 2021.
- [175] Martins Ezuma, Fatih Erden, Chethan K. Anjinappa, Ozgur Ozdemir, and Ismail Guvenc. Drone remote controller rf signal dataset, 2020. URL <https://dx.doi.org/10.21227/ss99-8d56>.
- [176] MHD Saria Allahham, Mohammad F Al-Sa’d, Abdulla Al-Ali, Amr Mohamed, Tamer Khattab, and Aiman Erbad. Dronerf dataset: A dataset of drones for rf-based detection, classification and identification. *Data in brief*, 26, 2019.
- [177] Martins Ezuma, Fatih Erden, Chethan Kumar Anjinappa, Ozgur Ozdemir, and Ismail Guvenc. Detection and classification of uavs using rf fingerprints in the presence of wi-fi and bluetooth interference. *IEEE Open Journal of the Communications Society*, 1:60–76, 2019.
- [178] Letizia Marchegiani and Paul Newman. Listening for sirens: Locating and classifying acoustic alarms in city scenes. *IEEE transactions on intelligent transportation systems*, 23(10):17087–17096, 2022.

-
- [179] Jiabao Yu, Aiqun Hu, Fen Zhou, Yuexiu Xing, Yi Yu, Guyue Li, and Linning Peng. Radio frequency fingerprint identification based on denoising autoencoders. In *2019 International Conference on Wireless and Mobile Computing, Networking and Communications (WiMob)*, pages 1–6. IEEE, 2019.
- [180] Xingcheng Ran, Yue Xi, Yonggang Lu, Xiangwen Wang, and Zhenyu Lu. Comprehensive survey on hierarchical clustering algorithms and the recent developments. *Artificial Intelligence Review*, 56(8):8219–8264, 2023.
- [181] Dhivya Chandrasekaran and Vijay Mago. Evolution of semantic similarity — a survey. *ACM Computing Surveys (CSUR)*, 54(2):1–37, 2021.
- [182] Hakan Aydın, Zeynep Orman, and Muhammed Ali Aydın. A long short-term memory (lstm)-based distributed denial of service (ddos) detection and defense system design in public cloud network environment. *Computers & Security*, 118:102725, 2022.
- [183] Davide Chicco. Siamese neural networks: An overview. *Artificial neural networks*, pages 73–94, 2021.
- [184] Manzil Zaheer, Guru Guruganesh, Kumar Avinava Dubey, Joshua Ainslie, Chris Alberti, Santiago Ontanon, Philip Pham, Anirudh Ravula, Qifan Wang, Li Yang, et al. Big bird: Transformers for longer sequences. *Advances in neural information processing systems*, 33:17283–17297, 2020.
- [185] Nesrine Kaaniche, Maryline Laurent, and Sana Belguith. Privacy enhancing technologies for solving the privacy-personalization paradox: Taxonomy and survey. *Journal of Network and Computer Applications*, 171:102807, 2020.