



University of Wales
Trinity Saint David
LONDON

A Defensive Cloud and Drone Security Posture Assessment Framework for Multi-Cloud UAV Operations

CD-SPAF: Cloud-Drone Security Posture Assessment Framework

A dissertation submitted in partial fulfilment of the requirements for the degree of MSc
Cyber Security and Digital Forensics

Author: Shahriar Badhan

Student ID: 2505521

Supervisor: Dr Dipo Dunsin

**University of Wales Trinity Saint David (UWTSD) London Campus
September 2026**

DECLARATION AND CONSENT

I declare that this dissertation represents my own work, except where otherwise stated, and that it has not been submitted, in whole or in part, for any other academic award at this or any other institution.

I confirm that all sources of information, including text, data, figures and ideas drawn from the work of others, have been properly acknowledged and referenced in accordance with the University's guidelines on academic integrity.

I confirm that the security assessment activities described in this dissertation were conducted within a controlled research environment using simulated or controlled data, and did not involve unauthorised access to, or testing of, third-party systems.

I consent to this dissertation being made available for consultation, and to it being copied in whole or in part for the purposes of study or research, subject to normal conditions of academic use and acknowledgement, and I consent to it being submitted to plagiarism-detection software as required by the University.

Name: Shahriar Badhan

Date: 25/9/2026

ABSTRACT

Unmanned aerial vehicles (UAVs) increasingly rely on cloud infrastructure for telemetry storage, processing, monitoring and fleet management, commonly through AWS or Microsoft Azure. This dependence widens the security boundary of a UAV operation beyond the aircraft itself to include cloud identity, storage, networking and monitoring, yet UAV security research and cloud security practice have largely developed separately, so security conditions in these two domains are typically assessed and reported independently. This dissertation addresses that gap by proposing, implementing and evaluating the Cloud-Drone Security Posture Assessment Framework (CD-SPAF), a defensive assessment framework that combines cloud security posture assessment, UAV telemetry anomaly detection, device identity assessment, security policy evaluation and risk scoring within a single workflow. Following a review of UAV cybersecurity, cloud security guidance and posture-assessment literature, the study adopts an applied, design-oriented methodology in which a controlled research prototype is developed in Python, using Isolation Forest and DBSCAN for telemetry anomaly detection, a YAML-based policy engine for security-rule evaluation, a multi-factor risk-scoring model, and a Streamlit dashboard for reporting. The prototype was evaluated using a controlled assessment scenario combining simulated UAV telemetry with representative cloud security conditions. The assessment run generated 36 security findings across five categories — UAV Telemetry, Cloud Storage, Cloud Identity, IoT Identity and Monitoring — with 94.4% classified as High severity and 5.56% as Medium, and every finding was mapped to a category-appropriate recommendation drawn from NIST CSF 2.0, the CIS AWS Benchmark, OWASP IoT guidance and MITRE ATT&CK Cloud. These results demonstrate that a unified UAV-cloud security posture assessment pipeline is technically feasible using standard, freely available tools, although they represent a controlled feasibility demonstration rather than a validated measure of real-world risk reduction. The dissertation concludes by identifying the framework's contribution, limitations and directions for future work, including live multi-cloud integration, real-time telemetry, and validation of the risk model against operational data.

KEYWORDS

UAV Security; Cloud Security Posture Management; Multi-Cloud Security; Drone Cybersecurity; Zero Trust Architecture; Risk Assessment; Policy-as-Code; Anomaly Detection; NIST Cybersecurity Framework; IoT Device Security

TABLE OF CONTENTS

CHAPTER 1 — INTRODUCTION	1
1.1 Background	1
1.2 Problem Statement.....	2
1.3 Research Aim	3
1.4 Research Objectives.....	3
1.5 Research Question	4
1.6 Scope of the Research.....	4
1.7 Research Contribution	5
1.8 Research Structure.....	5
CHAPTER 2 — LITERATURE REVIEW.....	6
2.1 UAV Systems and Security Requirements	6
2.2 UAV Communication and Telemetry Security	7
2.3 UAV Cybersecurity Threats	8
2.4 Cloud Computing in UAV Operations	9
2.5 AWS Security for Cloud-Connected UAV Systems	9
2.6 Azure Security for Cloud-Connected UAV Systems	10
2.7 Identity, IoT and Device Security	10
2.8 Cloud Security Posture Management	11
2.9 Zero Trust Architecture.....	11
2.10 NIST Cybersecurity Framework.....	12
2.11 Multi-Cloud Security	13
2.12 Security Assessment and Risk Scoring	13
2.13 Policy-as-Code for Security Assessment	14
2.14 Existing Security Assessment Approaches and Critical Comparison	14
2.15 Research Gap	15
2.16 Proposed Position of CD-SPAF within Existing Research	16
2.17 Chapter Summary	17
CHAPTER 3 — RESEARCH METHODOLOGY	18
3.1 Research Approach	18
3.2 Research Design	18
3.3 Research Environment and System Boundary.....	19
3.4 Data Sources and Data Collection.....	19
3.5 Threat Model and Security Assessment Strategy	20
3.6 CD-SPAF Framework Development Process	20

3.7 Prototype Implementation Method	21
3.8 Security Assessment Scenarios	22
3.9 Risk Scoring Method	22
3.10 Evaluation Metrics	22
3.11 Ethical and Legal Considerations	23
3.12 Methodological Limitations	23
3.13 Chapter Summary	24
CHAPTER 4 — CD-SPAF FRAMEWORK AND SYSTEM DESIGN	25
4.1 CD-SPAF Design Objectives	25
4.2 Overall CD-SPAF Architecture	25
4.3 Framework Components	26
4.3.1 Cloud Security Scanner	27
4.3.2 UAV Telemetry Analyzer	27
4.4 Security Policy Engine	27
4.5 Risk Assessment Engine	28
4.6 Security Recommendation Engine	28
4.7 CD-SPAF Assessment Workflow	29
4.8 Security Control Mapping	29
4.9 Data Flow within CD-SPAF	30
4.10 Design Considerations	30
4.11 Chapter Summary	31
CHAPTER 5 — IMPLEMENTATION	32
5.1 Implementation Environment	32
5.2 CD-SPAF Project Structure and Main Execution	33
5.3 UAV Telemetry Processing and Anomaly Detection	34
5.4 Security Policy Engine Implementation	36
5.5 Risk Assessment and Recommendation Engine Implementation	38
5.6 Dashboard Implementation	39
5.7 Integration of the CD-SPAF Components	40
5.8 Prototype Execution and Assessment Output	40
5.9 Security Recommendations Output	43
5.10 Implementation Security and Safety Considerations	44
5.11 Implementation Limitations	45
5.12 Chapter Summary	45
CHAPTER 6 — RESULTS AND ANALYSIS	46

6.1 Overview of Assessment Results	46
6.2 Security Finding Categories.....	46
6.3 Severity Distribution	47
6.4 Risk Score Analysis	48
6.5 Relationship Between Findings and Recommendations	49
6.6 Assessment Consistency	49
6.7 Chapter Summary	50
CHAPTER 7 — DISCUSSION	51
7.1 Interpretation of the Assessment Findings.....	51
7.2 Relationship Between UAV and Cloud Security.....	51
7.3 Telemetry Anomaly Detection in Context	52
7.4 Policy-Based Assessment and Risk Scoring in Context	52
7.5 Multi-Cloud Considerations	53
7.6 Achievement of Research Objectives	53
7.7 Answer to the Research Question	54
7.8 Practical Significance and Contribution	55
7.9 Limitations and Implications.....	55
CHAPTER 8 — CONCLUSION AND FUTURE WORK.....	57
8.1 Conclusion.....	57
8.2 Research Contribution	57
8.3 Research Limitations.....	58
8.4 Future Work.....	58
8.5 Researcher Reflection	59
REFERENCES.....	61

LIST OF FIGURES

Figure 1.1: Cloud-Connected UAV Data Flow, from Collection to Monitoring	2
Figure 2.1: Literature Domains Supporting the CD-SPAF Framework.....	6
Figure 3.1: CD-SPAF Research Methodology and Assessment Process.....	21
Figure 4.1: Overall CD-SPAF Architecture.....	26
Figure 4.2: CD-SPAF Security Policy Evaluation Process	28
Figure 4.3: CD-SPAF Security Assessment Workflow	29
Figure 5.1: CD-SPAF Project Structure and Main Execution	34
Figure 5.2: UAV Telemetry Anomaly Detection Implementation	35
Figure 5.3: CD-SPAF Security Policy Evaluation Implementation.....	37
Figure 5.4: Risk and Security Recommendation Mapping	38
Figure 5.5: Streamlit Dashboard Implementation	39
Figure 5.6: CD-SPAF Security Findings Dashboard	41
Figure 5.7: CD-SPAF Security Findings and Evidence	41
Figure 5.8: CD-SPAF Dashboard — Severity Distribution and Risk Score by Category	42
Figure 5.9: CD-SPAF Findings Table Extract — DBSCAN-Detected Telemetry Observations.....	42
Figure 5.10: CD-SPAF Security Recommendations Output (Extract).....	44

LIST OF TABLES

Table 2.1: Conceptual relationship between NIST CSF 2.0 and CD-SPAF	12
Table 2.2: Comparison of Existing Security Approaches with CD-SPAF	15
Table 4.1: CD-SPAF Security Control Mapping	30
Table 5.1: CD-SPAF Implementation Environment	33
Table 6.1: CD-SPAF Security Finding Categories	46
Table 6.2: Observed Distribution of Findings by Category	47
Table 6.3: Severity Distribution of Generated Findings.....	48

LIST OF ABBREVIATIONS

• API	Application Programming Interface
• AWS	Amazon Web Services
• CD-SPAF	Cloud-Drone Security Posture Assessment Framework
• CIS	Center for Internet Security
• CSA	Cloud Security Alliance
• CSF	Cybersecurity Framework
• CSPM	Cloud Security Posture Management
• CSV	Comma-Separated Values
• DBSCAN	Density-Based Spatial Clustering of Applications with Noise
• GNSS	Global Navigation Satellite System
• GPS	Global Positioning System
• IAM	Identity and Access Management
• IoT	Internet of Things
• JSON	JavaScript Object Notation
• MAVLink	Micro Air Vehicle Link
• MITRE ATT&CK	MITRE Adversarial Tactics, Techniques and Common Knowledge
• MQTT	Message Queuing Telemetry Transport
• NIST	National Institute of Standards and Technology
• OPA	Open Policy Agent
• OWASP	Open Worldwide Application Security Project
• S3	Amazon Simple Storage Service
• SIEM	Security Information and Event Management
• SITL	Software-in-the-Loop
• SOC	Security Operations Centre
• SP	Special Publication
• UAV	Unmanned Aerial Vehicle
• YAML	YAML Ain't Markup Language
• ZTA	Zero Trust Architecture

CHAPTER 1 — INTRODUCTION

1.1 Background

Unmanned aerial vehicles (UAVs) are increasingly used for surveillance, environmental monitoring, agriculture, infrastructure inspection, emergency response and logistics [1]. Their ability to gather information from locations that are difficult or unsafe for humans has made them valuable in both commercial and public-sector settings. Modern UAV systems, however, are rarely isolated flying devices: they depend on communication networks, ground control stations, cloud services, storage platforms and data-processing components to support their operation.

Cloud computing extends UAV capability by providing scalable storage, processing, monitoring and centralised management. Telemetry and other operational data generated in flight can be transferred to a cloud environment for storage and analysis, while cloud platforms can also support device identity management, logging, access control and security monitoring. This combination enables more flexible, fleet-scale UAV operations, but it also widens the security boundary: a problem affecting the UAV itself, a ground station, a communication channel, a cloud identity, a storage resource or a monitoring service can affect the rest of the system. Security assessment therefore cannot be confined to the aircraft alone.

UAV systems face threats that differ from those of conventional information systems, including communication interception, GNSS spoofing, jamming, denial-of-service conditions and malware directed at UAV networks [2]–[5]. Cloud-connected UAV systems add risks associated with identity and access management, insecure storage, excessive permissions, insufficient logging and weak device identities. A further complication is that a cloud-supported UAV architecture may draw on more than one provider — commonly AWS or Microsoft Azure — each with its own services, configuration mechanisms and security controls, which makes joint assessment of UAV and cloud conditions more difficult.

Structured cybersecurity risk management is well established at the standards level. The NIST Cybersecurity Framework (CSF) 2.0 offers a common approach for understanding, assessing,

prioritising and communicating cybersecurity risk through six functions — Govern, Identify, Protect, Detect, Respond and Recover [6]. NIST SP 800-207 describes Zero Trust Architecture as an approach in which trust is never granted implicitly on the basis of network location; authentication and authorisation are instead treated as continuous requirements [7]. These principles are directly relevant to cloud-connected UAV environments, where devices, ground stations and cloud resources routinely operate across different networks and administrative boundaries. Applying individual controls in isolation, however, does not by itself produce a unified view of the security condition of a UAV-cloud system, which is the gap this research addresses. Figure 1.1 summarises the simplified data flow assumed throughout this dissertation, from data collection on the UAV, through the ground or edge system, to cloud-based storage and security monitoring.

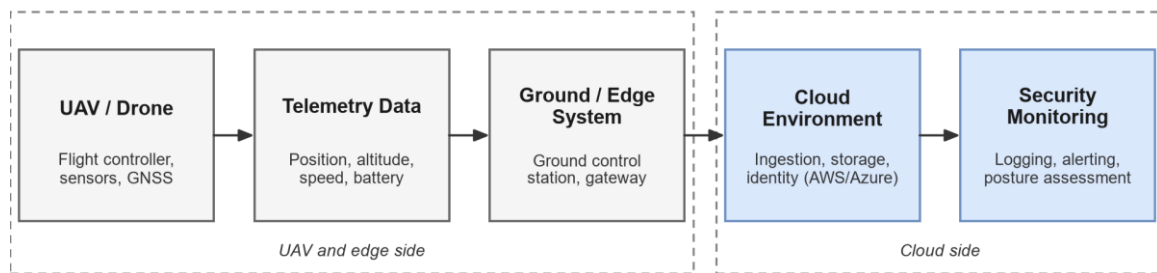


Figure 1.1: Cloud-Connected UAV Data Flow, from Collection to Monitoring

1.2 Problem Statement

Integrating UAV systems with cloud platforms brings clear operational benefits, but it also creates a distributed security environment in which security-relevant information exists across several layers: the UAV itself, telemetry data, the ground control station, the network, cloud identity, cloud storage and monitoring infrastructure.

A central problem is that these layers are typically assessed separately. A cloud security review may flag an insecure storage configuration or an identity weakness, while a UAV-focused review may flag abnormal telemetry or an unusual device identity. Considered independently, it is difficult to judge how these findings relate to the overall safety and security of the operation. The problem is compounded when more than one cloud provider is involved: AWS and Azure expose different services and configuration models, so an

assessment method built around a single provider does not transfer cleanly to a multi-cloud deployment.

This research addresses the resulting need for a structured, defensive framework that can collect security-relevant information from both UAV and cloud components, apply predefined security policies, identify findings, and present the results in a common risk-oriented format. The proposed Cloud-Drone Security Posture Assessment Framework (CD-SPAF) is defined accordingly as a defensive assessment and risk-scoring framework for identifying and helping to remediate cloud-drone security weaknesses within an authorised, controlled environment. It is not intended to guarantee that a UAV system is completely secure, nor to detect every possible attack; its purpose is to identify predefined security conditions and generate risk information and recommendations within the scope of the research prototype.

1.3 Research Aim

The aim of this research is to design, develop and evaluate a defensive security posture assessment framework for UAV operations that depend on cloud infrastructure across multiple cloud environments. CD-SPAF brings together security assessment of UAV telemetry, device identity and selected cloud configuration conditions into a common assessment process that generates findings, assesses risk and produces recommendations to support defensive security improvement. The research does not attempt to build a production-ready autonomous security system; the objective is to demonstrate and evaluate the feasibility of the assessment approach through a controlled prototype.

1.4 Research Objectives

The research is guided by the following objectives, which are consistent with those established in the original research proposal:

1. To identify security weaknesses that may occur within cloud-supported UAV architectures, including issues related to identity and access management, cloud storage, logging, device identity, network exposure and telemetry.
2. To design the CD-SPAF framework using relevant cybersecurity principles and established security guidance, including NIST CSF 2.0 and Zero Trust principles.

3. To develop a prototype capable of collecting selected cloud configuration information and UAV or controlled telemetry data and applying predefined security assessment rules.
4. To implement a risk assessment mechanism that categorises security findings according to their severity and relevance to the assessed UAV-cloud environment.
5. To evaluate the prototype using controlled security scenarios and determine whether the framework can identify predefined security conditions and generate useful recommendations.

1.5 Research Question

The research is guided by the following primary research question:

How can a unified security framework assess and reduce cloud, IoT, and UAV-related risks in drone systems that depend on AWS or Azure for telemetry, storage, command services, analytics, and monitoring?

This question is addressed through the design of CD-SPAF, its prototype implementation and controlled evaluation, and specifically considers whether a unified assessment process can provide a clearer view of security conditions than treating UAV and cloud findings independently.

1.6 Scope of the Research

The scope is limited to defensive security assessment of cloud-supported UAV environments and covers three areas: the UAV and edge environment (telemetry and device identity), the cloud environment (selected identity, storage and monitoring conditions), and the assessment layer, in which collected information is evaluated against predefined policies and risk logic. The architecture assumed by the research proposal consists of a UAV or simulated telemetry source, a ground or edge component, cloud services, and the CD-SPAF assessment layer, with AWS IoT Core, Azure IoT Hub or an MQTT test broker as possible ingestion routes, and AWS S3, CloudWatch and Azure Monitor as representative cloud security and monitoring services.

The project does not require a physical flying UAV. Controlled or simulated telemetry — for example from PX4 SITL, ArduPilot SITL, MAVLink logs or generated datasets — allows the experiments to be conducted safely and repeatably. The implementation is therefore evaluated as a research prototype rather than a production multi-cloud UAV deployment, and the research focuses on a defined set of security conditions — insecure cloud storage, excessive permissions, monitoring or logging weaknesses, device identity issues and abnormal telemetry — rather than attempting to cover every possible UAV or cloud attack.

1.7 Research Contribution

The primary contribution of this research is the design and prototype implementation of the Cloud-Drone Security Posture Assessment Framework (CD-SPAF), which combines three related assessment areas — cloud security posture, UAV/edge security, and telemetry and operational security — into a single workflow that collects information, applies security policy checks, assigns risk information and presents findings through a security dashboard.

A further contribution is the attempt to represent UAV-related and cloud-related findings within one common assessment process rather than as separate outputs, giving the researcher a unified view of which areas of a cloud-connected UAV deployment require attention. The framework is also intended as a foundation for future work, for example incorporating real-time cloud monitoring, live UAV telemetry, more advanced anomaly detection, automated remediation and broader multi-cloud coverage.

1.8 Research Structure

The remainder of this dissertation is organised into seven further chapters. Chapter 2 reviews UAV and cloud security literature and establishes the research gap addressed by CD-SPAF. Chapter 3 sets out the research methodology, system boundary, data sources and evaluation approach. Chapter 4 presents the design of the CD-SPAF architecture, its components and workflow. Chapter 5 describes the prototype implementation, including the technologies used and screenshots of the working system. Chapter 6 reports the results of the controlled assessment run. Chapter 7 discusses what those results mean in relation to the literature, the research question and the framework's practical value. Chapter 8 concludes the dissertation and identifies directions for future work.

CHAPTER 2 — LITERATURE REVIEW

The integration of UAVs with cloud computing has created a distributed technological environment in which flight systems, communication networks, ground stations, IoT components and cloud services operate together. UAV cybersecurity has traditionally been studied through individual areas such as wireless communication, navigation security, software vulnerabilities and privacy, while cloud security research has separately developed extensive approaches to identity management, configuration assessment, logging and monitoring. This chapter reviews the literature relevant to CD-SPAF across both areas, compares the capabilities and limitations of existing approaches, and identifies the research gap that motivates the proposed framework. Figure 2.1 summarises the four literature domains drawn on in this chapter and shows how they converge on the outputs that CD-SPAF is intended to produce.

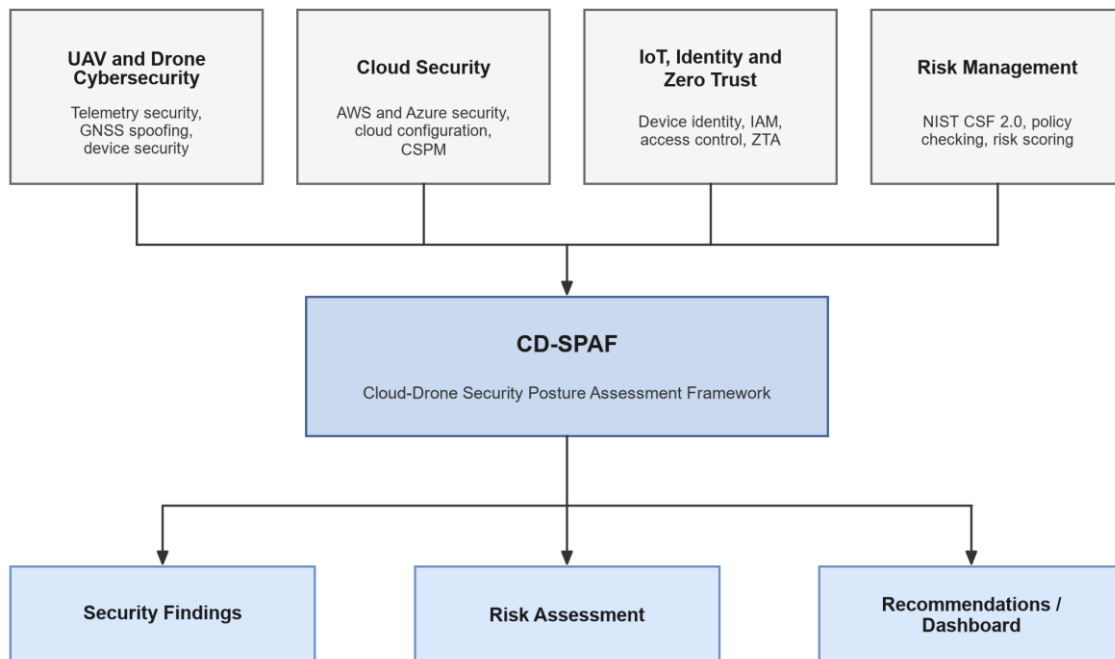


Figure 2.1: Literature Domains Supporting the CD-SPAF Framework

2.1 UAV Systems and Security Requirements

UAVs are cyber-physical systems in which flight operations depend on software, electronic components, communication technologies and data-processing mechanisms. A typical architecture includes a flight controller, navigation system, sensors, wireless interfaces and a

ground control station, often supplemented by cloud or edge services for storage, analytics, fleet management and remote monitoring. This combination of physical and digital components creates a broad security surface: a vulnerability in one component may affect others because the components are interconnected. A compromised communication channel could allow manipulation of information transmitted between a UAV and its ground station, and a compromised cloud identity could provide unauthorised access to UAV-generated data.

Shakhatreh et al. survey civil UAV applications and identify continued cybersecurity research as necessary as UAV use expands [1], while more recent surveys distinguish hardware, software and communication-technology attack surfaces and organise UAV security concerns around confidentiality, integrity and availability [5], [8]–[10]. Confidentiality matters because UAVs may collect sensitive operational or surveillance information; integrity matters because manipulated telemetry or control data can lead to incorrect decisions; and availability is critical because interrupted communication or supporting services can disrupt an ongoing mission. These characteristics distinguish UAV systems from many conventional information systems: assessment needs to consider not only whether data or systems are protected, but whether a given security condition could affect the operation of the aircraft itself. For CD-SPAF, this means the UAV is not treated as an isolated endpoint but as part of a wider system in which security-relevant information can originate from either the UAV side or the cloud side.

2.2 UAV Communication and Telemetry Security

Communication is fundamental to UAV operation because the aircraft typically exchanges telemetry — location, altitude, velocity, battery status, sensor readings and system state — with a ground control station or other systems. He, Chan and Guizani identify communication security as a critical UAV requirement and examine vulnerabilities associated with UAV modules and practical attacks involving GPS and Wi-Fi interfaces [2]. Related work on UAV and cellular-connected communications highlights the opportunities and open challenges created by increasingly wireless, networked UAV architectures [11], [12], and experimental studies of micro aerial vehicle networks demonstrate that these communication links are a realistic attack surface rather than a purely theoretical concern [13].

GNSS spoofing is one of the most consequential threats in this space: false navigation signals cause a receiver to compute an incorrect position or timing value. Psiaki and Humphreys examine spoofing methods and detection techniques in detail [14], and later work extends this to systematic reviews of GPS spoofing in flying ad hoc networks [15], machine-learning-based defences [16], onboard-sensor detection [17], and spoofing against multi-antenna drone swarms [18]; jamming methods aimed at UAV communication and navigation signals present a related but distinct threat [19]. GNSS spoofing is relevant to this research because it demonstrates why UAV security cannot rely on cloud configuration checks alone, but a framework focused only on navigation would not identify problems in cloud storage, identity management or logging. Telemetry can also be treated as a security-monitoring signal in its own right rather than only as operational data: unusual message frequency, unexpected connection failures, or significant deviation from expected values can be examined as potential findings requiring investigation. CD-SPAF's telemetry anomaly component is designed around this idea, using simulated telemetry and indicators such as abnormal message bursts, route deviation and repeated connection failures as assessment inputs, without attempting to replace specialised navigation-security systems.

2.3 UAV Cybersecurity Threats

UAV threats can be grouped into communication attacks, navigation attacks, software and device attacks, denial-of-service conditions, and identity-related attacks. Communication attacks target the wireless link between the UAV and ground system, aiming to intercept, manipulate or disrupt messages; because UAV operation often depends on continuous communication, availability is a particular concern [3], [4], [20], [21]. Encryption, authentication and integrity protection reduce this risk, but only if correctly configured and monitored — which is where a posture assessment layer can complement, rather than replace, dedicated communication security controls.

GNSS spoofing represents a navigation-specific instance of the same underlying problem: this research does not attempt to build a new detection algorithm, but uses GNSS-related anomalies to illustrate the wider need for UAV monitoring capable of recognising abnormal operational conditions. Denial-of-service conditions affecting a cloud API, communication broker or supporting service can also have operational consequences for UAV missions [22];

a posture assessment framework should distinguish between identifying a configuration that increases exposure to such attacks and actually detecting an active attack in progress, and CD-SPAF is designed for the former. Detection of attacks in progress is the concern of specialised UAV intrusion-detection work, such as the game-theoretic intrusion detection and ejection framework proposed by Sedjelmaci, Senouci and Ansari [23], which operates at the network level and assumes a very different evidence base from configuration posture assessment. Finally, UAVs depend on flight-control software, operating systems and communication applications whose vulnerabilities create additional risk, and on device identities — certificates or tokens used by cloud services to determine which device is communicating. If multiple devices share credentials, identifying the source of suspicious activity and revoking access when required both become harder, which is why device identity is treated as its own assessment area within CD-SPAF.

2.4 Cloud Computing in UAV Operations

Cloud computing extends UAV capability by providing scalable resources for storage, processing, monitoring and application hosting, reducing the processing burden on the UAV itself and making information accessible to authorised users from multiple locations. For fleet-based operations, centralised cloud services support management of many devices at once [22], [24]. This benefit comes with a corresponding cost: cloud integration expands the security boundary from the UAV and ground station alone to cloud identities, APIs, storage, networks, monitoring systems and data. A weakness at any one of these layers can affect the overall security posture, so the relevant question is not simply whether the cloud is secure, but how cloud resources are actually being used by UAV-related systems — the framing adopted throughout the remainder of this review.

2.5 AWS Security for Cloud-Connected UAV Systems

The AWS Well-Architected Security Pillar organises cloud security into seven areas: security foundations, identity and access management, detection, infrastructure protection, data protection, incident response and application security [25]. Several of these translate directly to a UAV context. AWS recommends establishing a strong identity foundation and applying least privilege, so a cloud service that receives UAV telemetry should not automatically hold broad permissions over unrelated resources. AWS also treats detection — distinguishing

unwanted configuration change from unexpected behaviour, supported by logging and centralised findings — as a core security function; the absence of expected monitoring or logging is therefore a legitimate CD-SPAF finding in its own right. Data-protection guidance covering encryption and access control at rest and in transit is directly relevant given that UAV telemetry can contain sensitive operational information, and infrastructure-protection guidance on defence in depth, trust boundaries and enforcement points confirms that AWS already provides extensive individual security mechanisms. CD-SPAF's role is not to replace these services but to provide a research framework for assessing selected conditions and combining them with UAV-side findings.

2.6 Azure Security for Cloud-Connected UAV Systems

Microsoft's Cloud Security Benchmark provides comparable recommendations for Azure workloads, data and services, covering identity management, privileged access, data protection, asset management, logging and threat detection, and explicitly addresses multi-cloud environments [26], [27]. Because AWS and Azure differ in services, terminology and configuration mechanisms, a framework intended to assess both needs some level of abstraction: a requirement such as "telemetry storage should not be publicly accessible" is represented differently by each provider's storage and permission model. A common policy layer can express the security requirement independently of the cloud provider, while provider-specific collectors translate it into the appropriate configuration check — the concept that directly underlies the policy-as-code component proposed for CD-SPAF, including requirements such as telemetry storage encryption and unique per-drone certificates.

2.7 Identity, IoT and Device Security

Cloud-connected UAV systems are closely related to IoT because a UAV behaves as a connected endpoint sending information to cloud services. Device identity is therefore an important requirement: unlike a human user, a UAV communicates automatically, and the cloud environment must be able to distinguish legitimate devices and apply appropriate permissions. A weak device-identity model creates concrete problems — if multiple devices share credentials, identifying the source of suspicious activity becomes harder, and if credentials are not revoked when a device is retired, access can remain available

unnecessarily. Both major providers treat device identity as a core IoT control: AWS IoT Core authenticates devices with X.509 certificates and authorises their actions through IoT policies [28], and Azure IoT solutions similarly rely on individual device identities and credentials [29]. The OWASP IoT Top 10 also places weak, guessable or hard-coded passwords and insecure default settings among the most common IoT weaknesses [30]. The original research proposal accordingly identified a drone/edge asset inventory (UAV identifier, firmware version, telemetry endpoint, certificates and data flows) as a useful input to cloud-drone assessment. It also marks an important difference between generic cloud posture assessment and cloud-drone posture assessment: a generic scanner may know a credential exists, but a UAV-aware assessment additionally needs to know which device that credential represents and what operational role the device performs, since the same configuration problem can carry different significance depending on the affected asset.

2.8 Cloud Security Posture Management

Cloud Security Posture Management (CSPM) focuses on identifying and managing configuration risk in environments where cloud resources change frequently, covering issues such as excessive permissions, insecure storage access, missing encryption, inadequate logging, exposed network services and insufficient monitoring [31]. AWS recommends automated, code-defined security practices and configuration-change detection as core detection capabilities, and Microsoft's benchmark similarly emphasises the difficulty of maintaining consistent security configuration across multiple cloud platforms [25]–[27]. CSPM is therefore highly relevant to CD-SPAF, but it generally stops at the cloud configuration itself: it may identify that a storage resource is publicly accessible without knowing whether that resource holds UAV telemetry or whether the associated UAV is behaving abnormally. This distinction can be summarised as two different questions — CSPM asks whether the cloud configuration is secure, while CD-SPAF asks what the security condition of the cloud-connected UAV environment is, considering cloud configuration and UAV-related information together. The second question is the central motivation for combining CSPM concepts with UAV security assessment.

2.9 Zero Trust Architecture

NIST SP 800-207 describes Zero Trust Architecture (ZTA) as an approach that moves security away from implicit trust based on network location and instead focuses explicitly on users, assets, resources, authentication and authorisation [7]. This is particularly relevant to UAV environments because a UAV may communicate with cloud services across different networks and should not be treated as trustworthy simply because it is connected to an expected network; its identity, authentication status, permissions and behaviour can all be considered before access is granted. NIST SP 800-207A extends these principles to cloud-native applications deployed across multi-location, including multi-cloud, environments, emphasising identity-based policy and access control across services [32], which is directly relevant to the multi-cloud dimension of CD-SPAF. Zero Trust is nevertheless not a complete replacement for posture assessment: it provides principles for access control and resource protection but does not itself perform telemetry analysis, cloud configuration checking, finding aggregation or risk-dashboard generation. CD-SPAF therefore treats Zero Trust as one source of security requirements while retaining separate mechanisms for posture assessment.

2.10 NIST Cybersecurity Framework

NIST CSF 2.0 is intended to help organisations understand, assess, prioritise and communicate cybersecurity risk through six functions: Govern, Identify, Protect, Detect, Respond and Recover [6]. The framework is useful for this research precisely because it provides a high-level structure rather than a specific technology, and each function can be related conceptually to part of the CD-SPAF assessment process, as summarised in Table 2.1.

Table 2.1: Conceptual relationship between NIST CSF 2.0 and CD-SPAF

NIST CSF 2.0 Function	Relevance to CD-SPAF
Govern	Define security requirements and assessment policies
Identify	Identify UAV, cloud and supporting assets
Protect	Assess access, storage and security controls
Detect	Identify abnormal telemetry and security findings
Respond	Provide recommended security actions
Recover	Support future remediation and security improvement

This mapping is conceptual rather than a claim of compliance: CD-SPAF does not implement the complete NIST CSF, and an MSc research prototype should not claim conformance with an established standard merely because some of its concepts have been reused. CSF 2.0 instead provides a useful organising vocabulary for the assessment process described in later chapters.

2.11 Multi-Cloud Security

Multi-cloud environments add management complexity because different providers expose different APIs, services, configuration structures and security mechanisms. Microsoft's benchmark explicitly identifies multi-cloud operation as a use case and discusses the difficulty of aligning security-control monitoring across platforms [26], and NIST SP 800-207A shows that Zero Trust principles extend to cloud-native applications deployed across multi-location and multi-cloud environments [32]. For UAV systems this complexity compounds further, since the cloud environment is only one part of a system that may also include an edge gateway, a cloud ingestion service and one or more storage or analytics services. If an insecure AWS storage configuration, an excessive Azure permission, an abnormal UAV telemetry reading and a certificate-identity issue are reported independently, a security team may struggle to see the overall picture. A common representation — recording each finding's affected asset, category, severity, likelihood, potential impact, affected layer and recommended action — is the approach adopted by CD-SPAF to address this problem.

2.12 Security Assessment and Risk Scoring

Posture assessment involves more than identifying whether a control is present: the importance of a finding depends on the asset affected and its possible consequences. A logging gap may matter because it reduces visibility, while an overly permissive identity may create a direct unauthorised-access risk, so significance is inherently context-dependent. NIST SP 800-30 treats risk as a function of the likelihood of a threat event and the adverse impact that would result, and expects organisations to tailor their assessment approach and scales to their own context [33]. Qualitative and semi-quantitative scales of this kind are widely used because reliable quantitative risk data are rarely available, but their outputs depend heavily on the assumptions made by whoever defines the scales. The CD-SPAF research proposal

suggested ranking findings by likelihood, impact, affected layer and remediation difficulty; in this research those considerations were condensed into a four-factor expression:

$$\text{Risk Score} = \text{Asset Criticality} \times \text{Exposure} \times \text{Control Failure} \times \text{Mission Impact}$$

This expression should be read as a research model rather than an industry-standard formula; its implementation and evaluation are described in Chapters 4 and 6. A further principle carried through the design is that a useful assessment should not treat every finding as an independent vulnerability — the prototype, for example, may generate several related telemetry findings during one run, which are more accurately reported as related security findings with an explained category and severity than as separate, unrelated vulnerabilities.

2.13 Policy-as-Code for Security Assessment

Policy-as-code expresses security requirements in a structured, machine-evaluable form rather than requiring every resource to be checked manually. A requirement such as "telemetry storage must use encryption", "each UAV must have a unique device identity" or "required security logging must be enabled" can be represented as a rule; the CD-SPAF proposal identifies YAML or Open Policy Agent (OPA) rules as suitable mechanisms, OPA being a general-purpose engine that evaluates declarative policies against structured input [34]. This is useful because it provides a common abstraction across cloud providers: a high-level policy expresses the requirement while provider-specific collectors obtain the relevant configuration data, cleanly separating collection (obtaining security-relevant information), policy evaluation (determining whether the information meets the requirement) and risk assessment (determining the significance of the resulting finding). This three-way separation underlies the modular architecture presented in Chapter 4. Policy-as-code is not without weaknesses, however: a rule set is only as good as the requirements it encodes, and an incomplete rule set can give a misleading impression of compliance simply because nothing failed.

2.14 Existing Security Assessment Approaches and Critical Comparison

The literature and official guidance reviewed above show that many individual security capabilities already exist. UAV research provides techniques for protecting communication and navigation; AWS and Azure provide extensive identity, detection, data-protection and

infrastructure-protection capabilities; and NIST provides general risk-management guidance through CSF 2.0 and identity-centred principles through Zero Trust. The research gap addressed by this dissertation is therefore not an absence of security tools, but the lack of an approach that integrates selected UAV, telemetry, cloud-configuration and identity findings into a single defensive assessment workflow for a cloud-connected UAV environment. Table 2.2 compares the scope of these existing approaches against the proposed CD-SPAF framework across five dimensions.

Table 2.2: Comparison of Existing Security Approaches with CD-SPAF

Approach	Main Purpose	UAV Awareness	Cloud Posture	Unified Risk View
UAV communication security research	Protect UAV communication	High	Low	Low
GNSS spoofing research	Detect navigation manipulation	High	Low	Low
UAV cybersecurity surveys	Identify broad UAV risks	High	Limited	Limited
AWS security guidance	Secure AWS workloads	Low	High	Provider-specific
Azure / MCSB guidance	Secure Azure and multi-cloud workloads	Low	High	Provider-oriented
CSPM approaches	Assess cloud configuration	Low	High	Moderate
Zero Trust Architecture	Protect resources via identity and access	Medium	High	Low
NIST CSF 2.0	Organise cybersecurity risk management	General	General	High-level
CD-SPAF (proposed)	Combined cloud-drone posture assessment	High	High	Yes

The table indicates that existing technologies provide important building blocks; the contribution of CD-SPAF is therefore combining these concepts into a common defensive workflow rather than inventing an entirely new security control.

2.15 Research Gap

Four related gaps emerge from this review. First, UAV security research generally focuses on the aircraft, communication system or navigation mechanism, whereas cloud security

approaches focus on cloud resources, and this separation makes it difficult to obtain a unified security view of a cloud-connected UAV system. Second, cloud posture assessment can identify configuration weaknesses but rarely incorporates UAV-specific operational context — a storage configuration may be technically insecure regardless of contents, but its operational significance differs depending on whether it holds UAV telemetry, mission data or non-sensitive test data. Third, AWS and Azure both provide extensive but structurally different security controls, and even Microsoft's own benchmark documentation recognises the challenge of maintaining consistent security monitoring across multiple cloud platforms [26]; a common assessment layer reduces the need to interpret every provider-specific finding independently. Fourth, existing approaches provide individual controls for identity, configuration, navigation, communication and monitoring, but these findings are not represented within a single risk-oriented model designed specifically for a cloud-connected UAV environment — this is the main gap addressed by CD-SPAF. The gap is therefore one of integration and emphasis rather than an absence of relevant prior work, and the contribution claimed for CD-SPAF is correspondingly modest.

2.16 Proposed Position of CD-SPAF within Existing Research

Based on this review, CD-SPAF is positioned as a defensive security assessment framework rather than as a replacement for AWS security services, Azure security services or specialised UAV intrusion-detection systems. It develops the five modules outlined in the original research proposal — a cloud posture scanner, a drone/edge asset inventory, a telemetry anomaly monitor, a policy-as-code engine and a risk dashboard — into six components: a Cloud Security Scanner, a UAV Telemetry Analyzer, a Security Policy Engine, a Risk Assessment Engine, a Security Recommendation Engine and a Dashboard/Reporting Layer. The asset-inventory idea is retained as device and asset information (for example, UAV identifiers and certificate status) that is evaluated by the policy engine rather than as a separate module, and recommendation mapping was separated from the dashboard so that guidance could be generated independently of its presentation. The framework therefore functions as an integration layer between existing security assessment sources. Its research value does not rest on a claim that existing AWS, Azure or UAV security controls are inadequate, but on investigating whether combining selected controls and assessment outputs can provide a

more coherent security-posture view for a cloud-connected UAV environment than the constituent parts provide separately.

2.17 Chapter Summary

This chapter reviewed UAV security requirements, communication and telemetry security, and the main categories of UAV cyber threat, before examining cloud computing's role in UAV operations and the security capabilities offered by AWS and Azure. CSPM, Zero Trust and NIST CSF 2.0 were considered as complementary but partial solutions, and multi-cloud security, risk scoring and policy-as-code were identified as concepts that CD-SPAF draws on directly. The review found no shortage of individual security technologies; rather, it identified a gap in integrating UAV, telemetry, cloud-configuration and identity findings into a single, risk-oriented assessment process for cloud-connected UAV operations. CD-SPAF is proposed to address this gap through six components — cloud security scanning, UAV telemetry analysis, policy-as-code evaluation, risk assessment, recommendation mapping and dashboard reporting — and the next chapter explains the methodology used to design, implement and evaluate the resulting framework.

CHAPTER 3 — RESEARCH METHODOLOGY

This chapter presents the methodology used to design, develop and evaluate CD-SPAF. It was developed to address the research question and objectives set out in Chapter 1 while maintaining a clear link between the security issues identified in the literature review and the implemented prototype. The research adopts a controlled, practical approach that uses simulated or controlled UAV-related telemetry together with cloud security information to create a repeatable assessment environment, rather than depending on a physical UAV deployment.

3.1 Research Approach

The research follows an applied, design-oriented cybersecurity approach: the main output is the CD-SPAF framework and its prototype, and the process moves from identifying a security problem, through analysing existing approaches and designing a framework, to implementing selected assessment functions and evaluating the resulting prototype. This combines concepts from UAV cybersecurity, cloud security, IoT security, posture assessment, risk management and security policy enforcement. Risk assessment specifically follows the general structure recommended by NIST SP 800-30 [33] for preparing, conducting and maintaining security risk assessments, adapted here to a cloud-connected UAV environment. The research does not attempt to build a complete commercial cloud security platform; its purpose is to demonstrate whether a unified framework can collect relevant security information, identify findings, assess risk, and produce useful recommendations.

3.2 Research Design

The research design consists of five connected stages: (1) literature and standards analysis, which identified security requirements and weaknesses associated with UAV systems, cloud platforms, IoT devices, identity management, telemetry and security monitoring; (2) framework design, which structured CD-SPAF around collecting security information from cloud and UAV-related components, followed by policy checking, risk assessment and reporting; (3) prototype development, in which the system was built to process controlled security information and telemetry-related data and generate structured findings; (4) controlled security assessment, covering scenarios such as cloud storage misconfiguration,

identity and access weaknesses, device or certificate identity issues, monitoring and logging gaps, and abnormal telemetry; and (5) evaluation and analysis of the findings produced, their risk classification, severity distribution and the usefulness of the generated recommendations. This provides a clear progression from theoretical research to practical implementation and evaluation.

3.3 Research Environment and System Boundary

The research was conducted within a controlled security assessment environment whose boundary includes UAV-related telemetry, cloud security configuration information, security policies, assessment logic, risk scoring and the CD-SPAF dashboard. The conceptual environment consists of four layers: the UAV or simulated UAV environment, which is the source of telemetry and device-related information; the ground or edge component, which receives and processes that information; the cloud environment, which provides the associated cloud services and security configuration; and the CD-SPAF assessment layer itself. AWS and Azure are considered as cloud environments within this context rather than as the research subject in their own right — the central subject is the CD-SPAF posture assessment framework. The implementation was deliberately kept within this controlled boundary: the research does not claim that a physical UAV was deployed in production, nor that a complete production-level AWS-Azure-UAV integration was achieved, which allows the assessment functions to be evaluated without the operational risk of a live UAV deployment.

3.4 Data Sources and Data Collection

Four categories of data support the assessment process. Academic literature and recognised standards — principally NIST CSF 2.0, NIST Zero Trust Architecture guidance and NIST SP 800-30 — provided the theoretical basis for the framework described in Chapter 2 [6], [7], [33]. Cloud and device security information was represented in the prototype by a small set of controlled posture conditions defined in the YAML policy file: a cloud storage condition (AWS-S3-001), a cloud identity condition (AWS-IAM-001), a device certificate condition (UAV-CERT-001) and a logging condition (AWS-LOG-001). These conditions were modelled on the types of misconfiguration addressed by AWS and CIS guidance [25], [35], but they were not collected from a live AWS or Azure account, and no credentials or personal data were involved. UAV telemetry was supplied as a controlled CSV dataset in which each record

contains a timestamp, drone identifier, latitude, longitude, altitude, speed and battery level; this provides operational information that a cloud posture assessment alone cannot capture. Finally, the prototype itself generates structured findings during an assessment run, which are written to a JSON report and subsequently used for severity classification, risk assessment, dashboard presentation and recommendation generation.

3.5 Threat Model and Security Assessment Strategy

Rather than attempting to model every possible attack against a UAV, the threat model concentrates on security posture conditions that the prototype can realistically assess: cloud identity and access management, cloud storage security, logging and monitoring, device identity and certificate management, UAV telemetry security, security policy compliance, abnormal telemetry-related conditions, and configuration weaknesses more generally. The relevant assets are the UAV, the ground or edge system, cloud services, identity mechanisms, communication paths and monitoring components. The assessment strategy is explicitly defensive: its purpose is to identify weaknesses and produce recommendations rather than to perform destructive attacks, consistent with the boundary of an authorised, controlled research environment.

3.6 CD-SPAF Framework Development Process

CD-SPAF was developed through five design stages. Security requirements — identity protection, secure storage, telemetry security, monitoring, device identity and policy compliance — were first derived from the literature review and identified problems in cloud-connected UAV systems, then organised into security domains so that different types of security information could be assessed within a common framework. Assessment rules were defined to convert collected information into structured findings, each of which is then evaluated for its security relevance and potential impact so that findings can be prioritised rather than left as an unorganised list. The final stage presents the results through the CD-SPAF dashboard, showing findings, risk information, severity distribution and recommended actions. This progression, illustrated in Figure 3.1, allows the framework to move from raw security information to actionable assessment output.

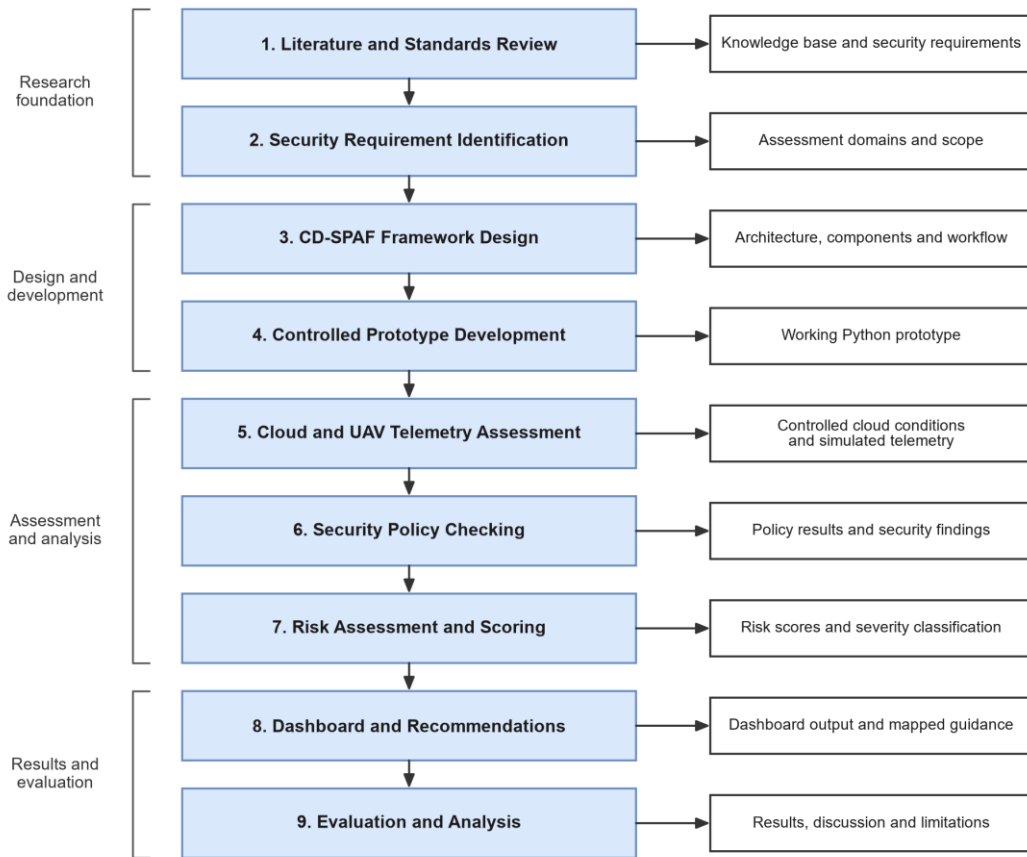


Figure 3.1: CD-SPAF Research Methodology and Assessment Process

3.7 Prototype Implementation Method

The prototype was implemented as a controlled security assessment application combining assessment logic, telemetry analysis, risk assessment and dashboard presentation. It uses programmatic processing to examine security information against predefined assessment conditions, with the resulting findings categorised by security area and severity, and includes functions corresponding to each main CD-SPAF component: cloud posture assessment, UAV/edge information assessment, telemetry analysis, security policy checking, risk assessment, dashboard reporting and recommendation generation. The implementation does not claim to replace native cloud security services; it provides a unified assessment layer that organises information from different areas into a common security-posture view, which is why the research focuses on the framework and assessment methodology rather than on building an alternative to AWS or Azure's own security services. In practical terms, an assessment run is started from the command line with `python main.py --demo`, which loads the telemetry dataset and the policy file, executes the analysis components and writes the results to `outputs/report.json`; the Streamlit dashboard is then launched separately and reads

that report. Separating execution from presentation in this way means that the same report can be re-examined without repeating the analysis.

3.8 Security Assessment Scenarios

Controlled assessment scenarios were used to evaluate whether CD-SPAF could identify different types of security condition, spanning five categories: cloud storage security (for example, inappropriate access settings); identity and access security (for example, excessive permissions); UAV device identity, given that UAV systems depend on trusted device identities to communicate with cloud services; monitoring and logging, since insufficient monitoring reduces the ability to detect suspicious activity; and UAV telemetry, where controlled data is analysed for abnormal behaviour. These scenarios represent different security domains rather than a single vulnerability type, which supports CD-SPAF's core objective of unified posture assessment.

3.9 Risk Scoring Method

Risk scoring prioritises the findings generated by the prototype, since an assessment that reports findings with no indication of relative importance offers limited practical value. The model considers four factors:

$$\text{Risk Score} = \text{Asset Criticality} \times \text{Exposure} \times \text{Control Failure} \times \text{Mission Impact}$$

Asset criticality reflects the importance of the affected asset or service; exposure reflects the extent to which the asset or condition is exposed to a threat; control failure reflects the extent to which an expected control is missing, weak or misconfigured; and mission impact reflects the possible effect on UAV-related operations. The resulting score supports a severity classification of Low, Medium, High or Critical. It is a research-specific risk scoring model: the factor values and severity thresholds are defined within the prototype for the purposes of this research, and the model is not presented as a universal industry-standard formula — it provides a consistent basis for comparing findings within the controlled research environment, informed by the general risk-assessment principles in NIST SP 800-30 regarding the identification and analysis of threats, vulnerabilities, likelihood and impact [33].

3.10 Evaluation Metrics

The prototype was evaluated against six observable criteria. Security findings were recorded and grouped by domain, with a deliberate distinction maintained between a finding — a condition identified by the assessment rules — and a confirmed, exploitable vulnerability. Severity distribution across findings indicates whether the risk engine provides meaningful prioritisation, while the risk scores themselves indicate whether different conditions receive different priority levels. The recommendations produced were examined for whether they provide practical remediation guidance, and repeated assessment runs using the same controlled inputs were expected to produce consistent findings and classifications, providing an indication of repeatability. Finally, the evaluation considered whether the prototype could combine UAV-related and cloud-related domains into a single assessment view. Detection-accuracy measures such as precision, recall, false-positive rate or F1 score were not calculated, because the controlled telemetry dataset was not labelled with ground truth; the evaluation is therefore a functional one rather than a measurement of detection performance.

3.11 Ethical and Legal Considerations

Because cybersecurity research can potentially affect systems, devices, networks and data belonging to other parties, this research follows a defensive and controlled approach throughout. Assessment activity was restricted to an authorised research environment and controlled data, and did not involve unauthorised access, exploitation of third-party systems, destructive testing or intentional disruption of operational UAV services. Using simulated or controlled telemetry also removed the safety risks associated with operating a physical UAV during security experiments. Where cloud security information was used, only information required for the assessment was considered, with sensitive credentials, personal information or unnecessary operational detail excluded from the research dataset. Findings generated by the prototype are reported as assessment results rather than being automatically presented as confirmed real-world vulnerabilities.

3.12 Methodological Limitations

Several limitations should be acknowledged. The research uses a controlled prototype environment rather than a complete production UAV deployment, so results may not capture every condition that could arise during real-world operations. The prototype does not

reproduce every AWS or Azure security capability; cloud platforms offer extensive services and configuration options, while CD-SPAF focuses on selected posture areas relevant to the research objectives. Simulated telemetry cannot fully reproduce the complexity of real flight — environmental conditions, communication interference, hardware limitations and physical attacks introduce variables that a controlled dataset does not capture. The risk-scoring model is specific to this research and, although informed by established risk-assessment principles, its numerical output should not be read as a universal industry risk rating. Finally, the number of assessment scenarios is limited by the scope and resources available to an MSc project; future research could evaluate CD-SPAF across larger multi-cloud environments, additional UAV platforms, larger telemetry datasets and a wider range of security scenarios.

3.13 Chapter Summary

This chapter set out an applied, design-oriented methodology combining literature and standards analysis with practical prototype development and controlled security assessment. It defined the system boundary, data sources, threat model, development process, implementation method, assessment scenarios, risk-scoring approach and evaluation metrics, with particular attention to distinguishing security findings produced by the prototype from confirmed real-world vulnerabilities. The controlled research environment provides an appropriate boundary for evaluating cloud and UAV-related security conditions without requiring a production UAV deployment. The next chapter presents the detailed design of CD-SPAF, including its architecture, modules, assessment workflow, risk engine and dashboard.

CHAPTER 4 — CD-SPAF FRAMEWORK AND SYSTEM DESIGN

This chapter presents the design of CD-SPAF, based on the security requirements, threats and research gaps identified in Chapters 2 and 3. Its purpose is to bring together UAV telemetry, device identity, cloud configuration, security policy and monitoring information within a single assessment process, rather than treating UAV and cloud security as separate concerns. The framework is designed as a controlled prototype, so the emphasis throughout is on demonstrating the assessment process and the integration of security information rather than reproducing every capability available in commercial cloud platforms.

4.1 CD-SPAF Design Objectives

Five objectives, derived from the research question and objectives in Chapter 1, guided the design. The framework should provide a unified security assessment view, since UAV telemetry, cloud security information, device identity and policy results can represent different parts of the same operational environment. It should support repeatable, automated posture assessment rather than a one-off manual review, so that the same checks could later be scheduled or run continuously, and support risk prioritisation, since findings vary in importance and need to be distinguished by severity and risk score. It should provide actionable recommendations rather than stopping at detection, associating each finding with relevant security guidance and remediation information. Finally, the framework should remain modular, so that individual components can be developed and tested separately while still contributing to the overall assessment process.

4.2 Overall CD-SPAF Architecture

As shown in Figure 4.1, CD-SPAF sits above the operational UAV-cloud environment and processes the information it produces through a sequence of information collection, security assessment, risk analysis and reporting:

*UAV → Telemetry/Device Information → Ground or Edge System → Cloud
Environment → CD-SPAF Assessment Layer → Security Findings → Risk
Assessment → Recommendations and Dashboard*

The cloud environment represents the cloud services or cloud-related security information associated with the UAV operation; AWS and Azure are both considered within this

architecture, but the framework itself does not depend on a specific provider, since the research addresses the common assessment problem rather than a single platform's configuration model. The CD-SPAF assessment layer receives relevant security information and applies the policy and risk assessment functions described below before the results are presented through the dashboard, supporting the central research idea of combining cloud and UAV security information into one posture assessment process. Within the assessment layer, the telemetry analyzer and the cloud security scanner work on different inputs, but both produce findings in the same structure, which are then scored by the risk assessment engine, mapped to guidance by the recommendation engine and presented through the dashboard.

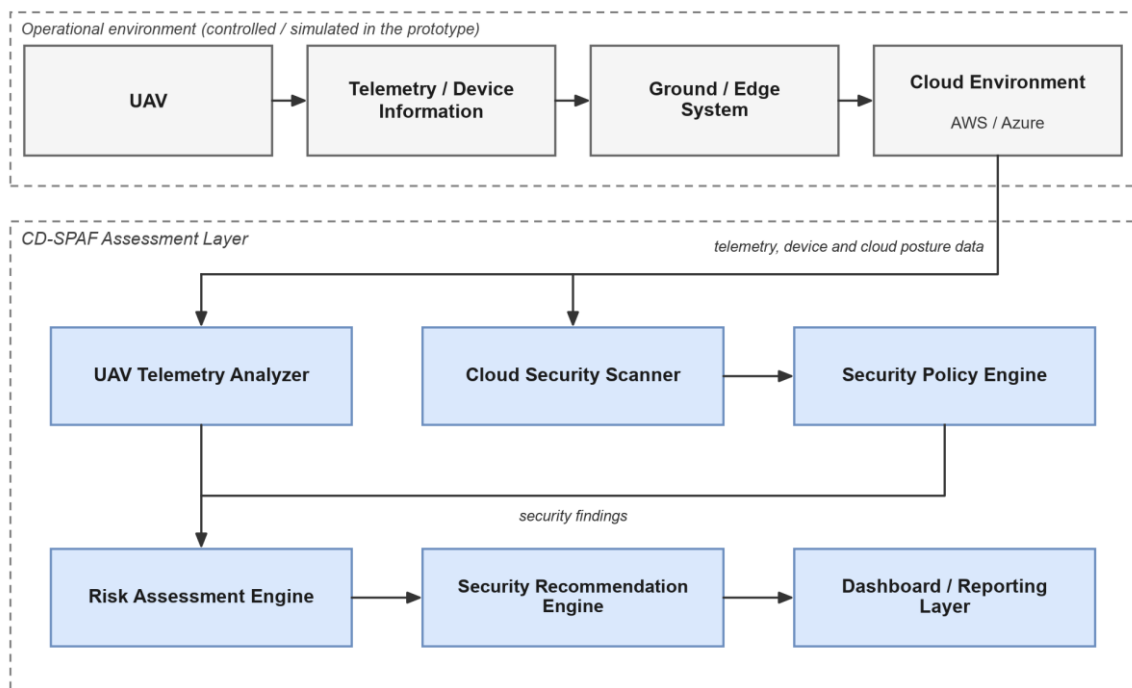


Figure 4.1: Overall CD-SPAF Architecture

4.3 Framework Components

CD-SPAF consists of six components: the Cloud Security Scanner, the UAV Telemetry Analyzer, the Security Policy Engine, the Risk Assessment Engine, the Security Recommendation Engine and the Dashboard/Reporting Layer. The first two are input-facing and deal with different kinds of evidence; the policy engine turns posture information into findings; and the remaining three are shared by every finding regardless of where it originated. The input-facing components are described below and the shared components in sections 4.4 to 4.6.

The Dashboard/Reporting Layer is a presentation component that reads the structured assessment report and displays findings, severity and recommendations; its implementation is described in section 5.6.

4.3.1 Cloud Security Scanner

The cloud security scanner examines security-relevant cloud information to identify conditions that may indicate weaknesses in storage, identity, access control or monitoring. It does not attempt to reproduce the full functionality of native cloud security services; instead it focuses on the selected security conditions relevant to the research objectives — for example, an inappropriate storage configuration or an identity and access control condition that requires review. In the prototype, this component is represented by a set of controlled cloud and device posture conditions rather than by live API collection from AWS or Azure; provider-specific collectors are identified as future work in Chapter 8.

4.3.2 UAV Telemetry Analyzer

The telemetry analyzer examines UAV-related telemetry, loading data from CSV input and processing numerical attributes such as latitude, longitude, altitude, speed and battery level. It applies machine-learning-based anomaly detection — Isolation Forest and DBSCAN — to identify unusual observations within the controlled dataset. The purpose is not to claim that every detected anomaly represents a cyberattack; an anomaly is treated as a security finding requiring further investigation, since abnormal telemetry may result from legitimate operational conditions as well as malicious activity.

4.4 Security Policy Engine

The policy engine provides the policy-evaluation layer of CD-SPAF using a YAML-based security policy file. It loads the defined policies and evaluates them against the available assessment information, providing a structured way of describing conditions that should be present or absent within the environment, and connecting predefined security requirements to automated, consistent assessment rather than manual, case-by-case checking. As shown in Figure 4.2, each rule is evaluated against the relevant security data: a satisfied condition is recorded as compliant, while an unsatisfied condition produces a finding that is passed to risk assessment:

*Security Data → Policy Rules → Policy Evaluation → Compliant / Finding →
Risk Assessment*

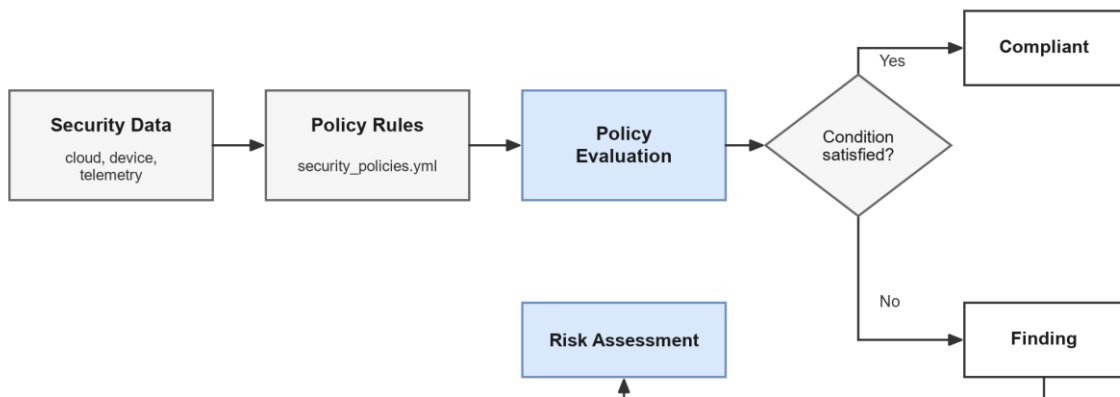


Figure 4.2: CD-SPAF Security Policy Evaluation Process

4.5 Risk Assessment Engine

The risk assessment engine assigns risk information to generated findings so that they are not presented as an unprioritised list, since different findings affect different assets and carry different levels of exposure and operational importance. As introduced in Chapter 3, the engine uses:

$$\text{Risk Score} = \text{Asset Criticality} \times \text{Exposure} \times \text{Control Failure} \times \text{Mission Impact}$$

The resulting score is associated with a Low, Medium, High or Critical severity classification. This scoring mechanism is specific to the research prototype rather than a universal industry-standard formula, and its role is to provide a consistent basis for comparing findings within the controlled assessment environment, broadly consistent with the risk-management structure of NIST CSF 2.0 [6] and the likelihood-and-impact reasoning of NIST SP 800-30 [33].

4.6 Security Recommendation Engine

The recommendation engine converts findings into remediation-oriented information by mapping security categories to recognised guidance, including NIST CSF 2.0, OWASP IoT guidance, the CIS AWS Benchmark and MITRE ATT&CK Cloud [6], [30], [35], [36]. A cloud storage finding, for example, may be mapped to relevant controls and a remediation suggestion, while a telemetry anomaly can be associated with monitoring-related guidance. This provides an important link between detection and practical security improvement: the

recommendations are presented as security guidance for review by a practitioner, not as an automatic claim that a specific attack has occurred.

4.7 CD-SPAF Assessment Workflow

The complete workflow proceeds through eight stages: data collection (obtaining available cloud configuration information and UAV-related telemetry or device information); data processing (converting the collected information into an analysable format); security policy evaluation; telemetry analysis; finding generation, in which conditions requiring attention are converted into structured findings; risk assessment, in which each finding is assigned a score and severity; recommendation mapping; and dashboard reporting, in which the final information is presented to the user. This is illustrated in Figure 4.3 and can be summarised as:

*Collect Security Information → Analyse Telemetry → Evaluate Policies →
Generate Findings → Calculate Risk → Map Recommendations → Display
Dashboard*

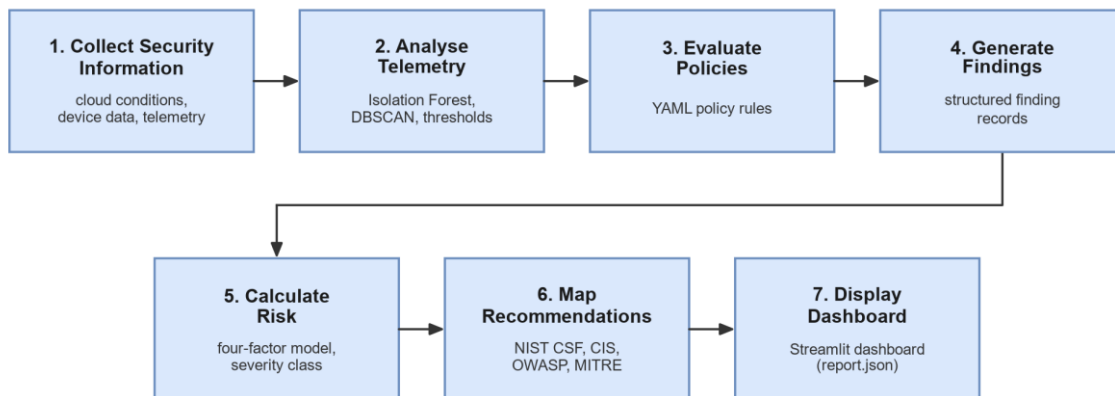


Figure 4.3: CD-SPAF Security Assessment Workflow

4.8 Security Control Mapping

CD-SPAF uses recognised frameworks and guidance as references for its assessment and recommendation process. NIST CSF 2.0 provides high-level cybersecurity outcomes [6] intended to help organisations understand and manage risk, and Zero Trust principles from NIST SP 800-207 [7] are used as supporting references for identity- and device-related security

rather than being implemented as a complete Zero Trust architecture. Table 4.1 summarises how each security area maps to CD-SPAF's assessment focus and example guidance.

Table 4.1: CD-SPAF Security Control Mapping

Security Area	CD-SPAF Assessment Focus	Example Guidance
Cloud Storage	Access and protection conditions	NIST CSF, CIS
Cloud Identity	Identity and access conditions	NIST CSF, Zero Trust
IoT / Device Identity	Device and certificate conditions	OWASP IoT
Monitoring	Logging and anomaly detection	NIST CSF, MITRE ATT&CK
UAV Telemetry	Abnormal telemetry observations	NIST CSF, OWASP IoT
Policy Compliance	Security policy conditions	NIST CSF, cloud benchmarks

4.9 Data Flow within CD-SPAF

The data flow begins with UAV-related and cloud-related information: UAV telemetry provides operational values such as position, altitude, speed and battery level, while cloud-related information provides posture information for areas such as storage, identity and monitoring. These inputs are processed separately where appropriate and combined at the assessment layer, where policy evaluation and risk assessment are applied. The final output is a structured assessment report containing findings, evidence, severity, risk scores and recommendations — a design that preserves the detail of each original finding while still providing a higher-level security posture view.

4.10 Design Considerations

Five considerations shaped the design. Modularity allows each major security function to be developed and evaluated independently; repeatability means the assessment process operates on controlled input data so that similar conditions can be assessed repeatedly; extensibility allows additional checks and cloud environments to be incorporated in future development; and the framework is deliberately defensive in operation, focused on assessment, detection, risk analysis and recommendations rather than destructive exploitation. Practicality also shaped the scope of the prototype: it was designed to remain achievable within the resources of an MSc project, which is one reason controlled and simulated UAV-related data are used rather than a full physical UAV deployment.

4.11 Chapter Summary

This chapter presented a CD-SPAF design that combines cloud security assessment, UAV telemetry analysis, security policy evaluation, risk scoring, recommendation mapping and dashboard reporting within a single framework. The architecture separates these functions while allowing their outputs to be combined into a common security posture view, and maintains a clear boundary between the research prototype and a production security platform: the goal is to demonstrate a practical assessment method rather than to replace native AWS or Azure security capabilities. The next chapter describes how this design was implemented, including the software environment, project structure and the evidence obtained from the working prototype.

CHAPTER 5 — IMPLEMENTATION

This chapter presents the practical implementation of CD-SPAF, converting the conceptual design in Chapter 4 into a working prototype capable of processing UAV-related telemetry and cloud security information, evaluating security conditions, generating findings, assigning risk information and presenting results through a dashboard. The implementation was developed as a controlled prototype rather than a production-ready UAV security platform: the objective is to demonstrate the feasibility of the assessment approach, not to deploy a complete operational security system for a real UAV fleet, and the prototype accordingly uses controlled security information and UAV telemetry data throughout. It follows the modular architecture introduced in Chapter 4, with a main execution module, a telemetry analysis module, a security policy engine, a risk assessment component, a recommendation engine, a report generation component and a Streamlit dashboard working together as:

*Security Information → Telemetry Analysis → Policy Evaluation → Finding
Generation → Risk Assessment → Recommendation Generation → Report
Generation → Dashboard*

5.1 Implementation Environment

The prototype was implemented in Python, chosen for its suitability for data processing, security analysis, machine-learning-based anomaly detection, rule-based policy evaluation and dashboard development. Pandas and NumPy handle telemetry processing; scikit-learn provides the Isolation Forest and DBSCAN techniques used by the telemetry component [37]–[39]; YAML stores security policies in a structured configuration format; and Streamlit provides the dashboard, presenting tables and metrics through a Python-based web application, with charts produced using Plotly Express. Table 5.1 summarises the resulting technology stack.

Table 5.1: CD-SPAF Implementation Environment

Component	Technology / Method	Purpose
Main application	Python	Coordinates the CD-SPAF modules
Telemetry processing	Pandas / NumPy	Loads and processes UAV telemetry
Anomaly detection	Isolation Forest / DBSCAN	Identifies unusual telemetry patterns
Policy evaluation	Python / YAML	Evaluates security conditions
Risk assessment	Python	Calculates and categorises risk
Recommendation engine	Python	Maps findings to security guidance
Reporting	JSON / Python	Stores structured assessment results
Dashboard	Streamlit / Plotly Express	Presents findings and risk information

The implementation is organised into separate modules rather than a single script, so that the telemetry detector, policy engine, risk engine and recommendation logic can each be understood, tested and extended independently, with the main application acting as the integration point. This is consistent with the research objective of building a framework rather than a single-purpose script, and makes it possible to replace or improve individual components in future work without redesigning the whole system.

5.2 CD-SPAF Project Structure and Main Execution

The main execution component is implemented in `main.py`, which acts as the central integration point for the CD-SPAF modules, as shown in Figure 5.1.

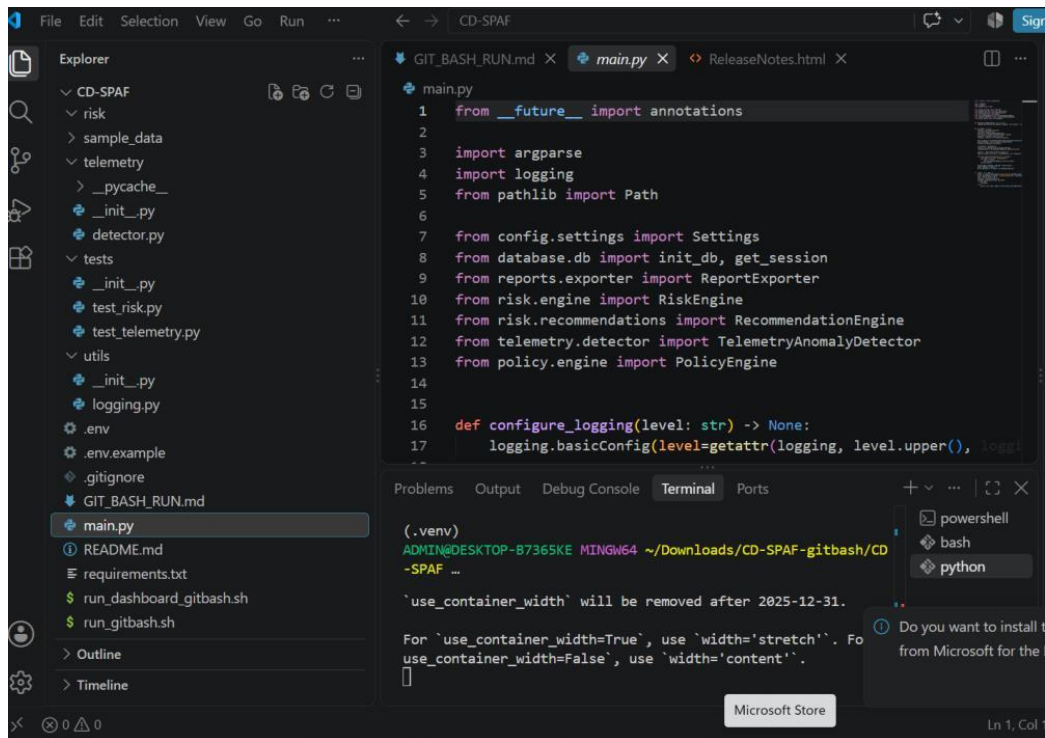


Figure 5.1: CD-SPAF Project Structure and Main Execution

Figure 5.1 shows the main application importing the major functional components required by the framework — configuration settings, database functions, the report exporter, the risk engine, the recommendation engine, the telemetry anomaly detector and the policy engine — so that cloud security, UAV telemetry, risk assessment and recommendations are connected into one defined assessment sequence rather than treated as independent activities. The process follows four broad stages: obtaining or loading the required security information; processing it through the relevant analysis components; generating findings and passing them to the risk and recommendation components; and exporting the resulting information for the dashboard. Separating configuration from processing logic in this way keeps the main execution file simple and the overall implementation easier to inspect, so the project structure itself reflects one of the main implementation decisions behind CD-SPAF: a collection of smaller components corresponding to the framework's conceptual functions, rather than one large program. The project also contains a small tests package (test_risk.py and test_telemetry.py) for the risk and telemetry modules, and a --demo command-line option that runs the controlled assessment scenario used in this dissertation.

5.3 UAV Telemetry Processing and Anomaly Detection

UAV telemetry is an important CD-SPAF input because security assessment for UAV operations cannot rely on cloud configuration alone; unusual telemetry patterns can provide indicators for further investigation. The TelemetryAnomalyDetector component, shown in Figure 5.2, uses Pandas and NumPy for data processing together with scikit-learn's machine-learning techniques.

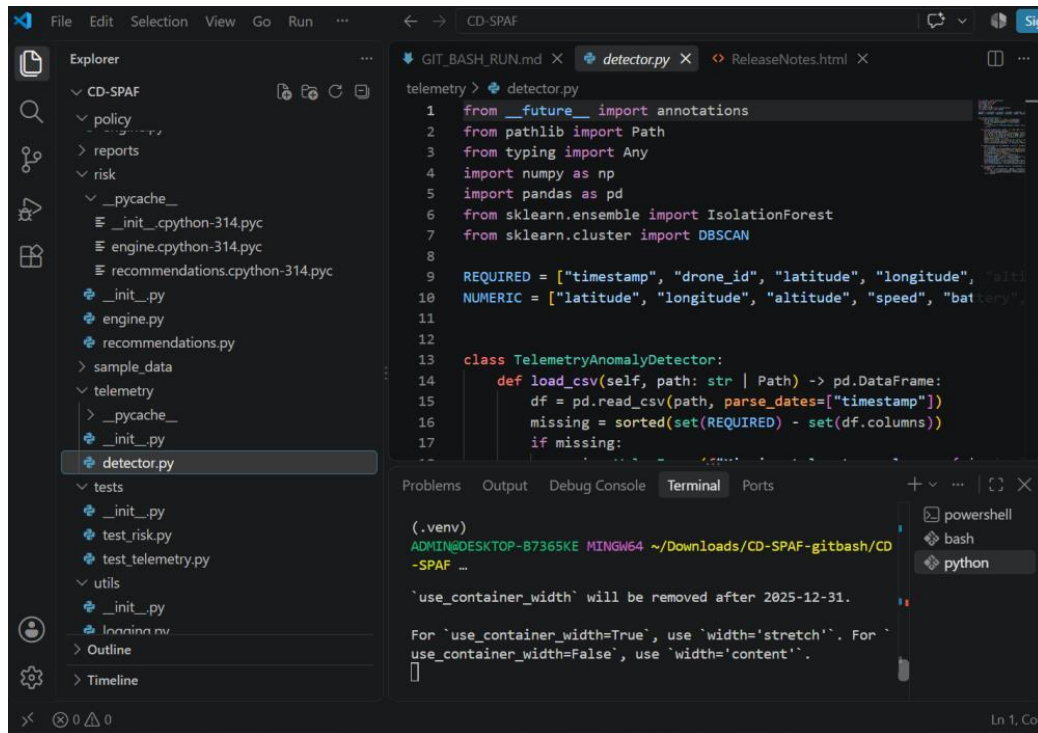


Figure 5.2: UAV Telemetry Anomaly Detection Implementation

The telemetry dataset used by the implementation contains a timestamp, drone ID, latitude, longitude, altitude, speed and battery status for each observation — a simplified set of fields representative of a real UAV telemetry source. The timestamp provides temporal context, the drone identifier associates observations with a specific UAV, the position fields describe location, altitude and speed describe flight behaviour, and battery status provides an additional operational indicator. The detector first loads the data and processes the timestamp to give a consistent temporal representation before examining the remaining attributes for unusual combinations.

Two complementary techniques are used. Isolation Forest identifies observations that can be isolated relatively easily from the rest of the dataset, and is applied here to flag telemetry observations that differ from the normal pattern in the input data [38]. DBSCAN instead groups observations by density and identifies points that do not belong to an established

cluster, which is useful where telemetry naturally forms groups based on operational characteristics [39]. Using both methods provides two different perspectives on unusual observations, but neither output is treated as automatic proof of a cyberattack: an unusual altitude, speed, position or battery reading may have a legitimate operational explanation — for example, an intentional change of altitude or speed as part of a mission — so CD-SPAF treats an anomaly as an assessment finding or indicator requiring investigation rather than a confirmed malicious event. This reduces the risk of over-interpreting machine-learning output and keeps the implementation aligned with the purpose of posture assessment rather than automated attack attribution; the component contributes mainly to the Detect function of the wider cybersecurity process described in NIST CSF 2.0 [6].

In the demonstrated run, the telemetry component produced 32 findings, identified in the output by the prefixes TEL-ISO (Isolation Forest, 10 findings), TEL-DBS (DBSCAN, 20 findings) and TEL-THR (2 findings). The TEL-THR findings come from a rule-based threshold check applied alongside the two learning-based detectors; the code for that check is not visible in the extract shown in Figure 5.2.

5.4 Security Policy Engine Implementation

The security policy engine provides CD-SPAF's rule-based assessment capability. Where the telemetry detector focuses on unusual UAV behaviour, the policy engine evaluates whether defined security conditions have been satisfied, using a PolicyEngine component that loads rules from a YAML configuration file named `security_policies.yml`, as shown in Figure 5.3.

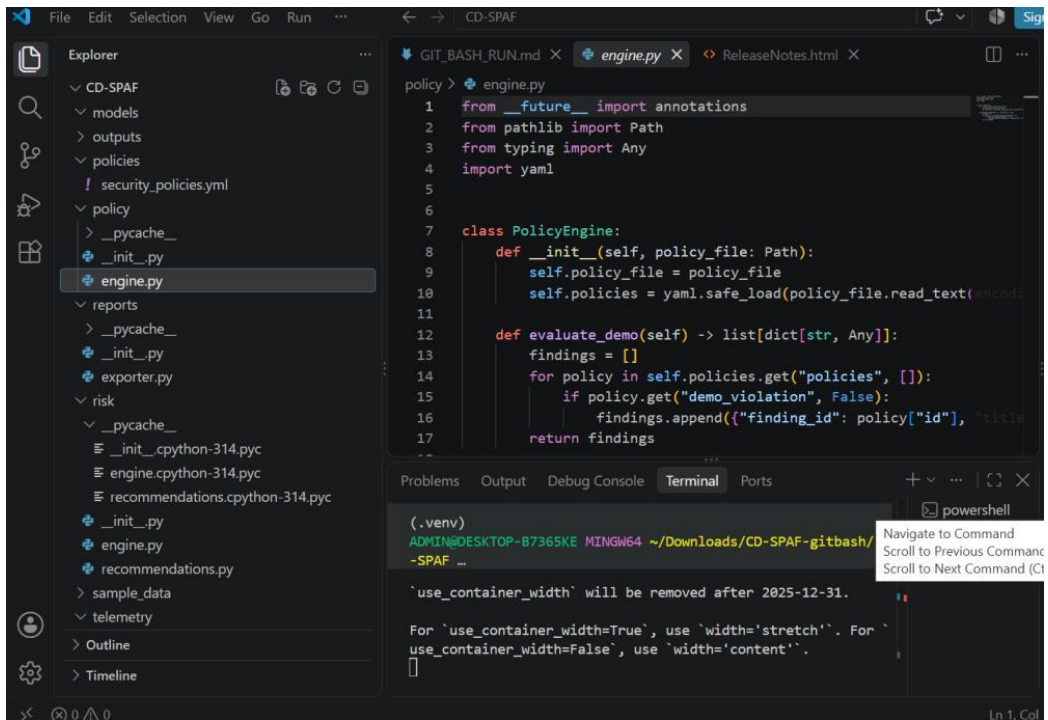


Figure 5.3: CD-SPAF Security Policy Evaluation Implementation

Storing rules in an external file allows them to be modified, and new rules to be introduced, without rewriting the application. The engine first receives the available security information — cloud configuration, UAV-related information, device identity information or monitoring-related information — then loads the relevant policy conditions and evaluates the information against them. If a condition is satisfied, the corresponding requirement is treated as compliant within the scope of the rule; if not, the engine generates a finding describing the identified condition, which is then passed to the risk engine. This converts security requirements into repeatable checks rather than relying on the consistency of a manual reviewer, and should be understood as a prototype policy-as-code mechanism rather than a complete enterprise compliance platform — its purpose here is to demonstrate how security requirements can be represented as machine-evaluable rules within the CD-SPAF architecture. In the demonstration configuration used for this dissertation, evaluation is performed by the engine's `evaluate_demo` method (Figure 5.3): each policy entry in `security_policies.yml` that is marked as a demonstration violation produces a finding carrying the policy's identifier and title. This is the route by which the four controlled cloud and device conditions described in section 3.4 entered the assessment. The mechanism demonstrates the path from a machine-readable rule to a structured finding, but it does not yet evaluate live configuration data retrieved from a cloud provider.

5.5 Risk Assessment and Recommendation Engine Implementation

Once the policy and telemetry components generate findings, CD-SPAF applies its risk assessment and recommendation logic. The risk component assigns a value to each identified condition using the model introduced in Chapter 3:

$$\text{Risk Score} = \text{Asset Criticality} \times \text{Exposure} \times \text{Control Failure} \times \text{Mission Impact}$$

Asset criticality reflects the affected asset's importance to the UAV or cloud environment; exposure reflects the degree to which the condition may be exposed to a threat; control failure reflects the absence or weakness of a relevant control; and mission impact reflects the potential effect on the UAV operation or associated cloud service. The resulting value supports severity classification and prioritisation.

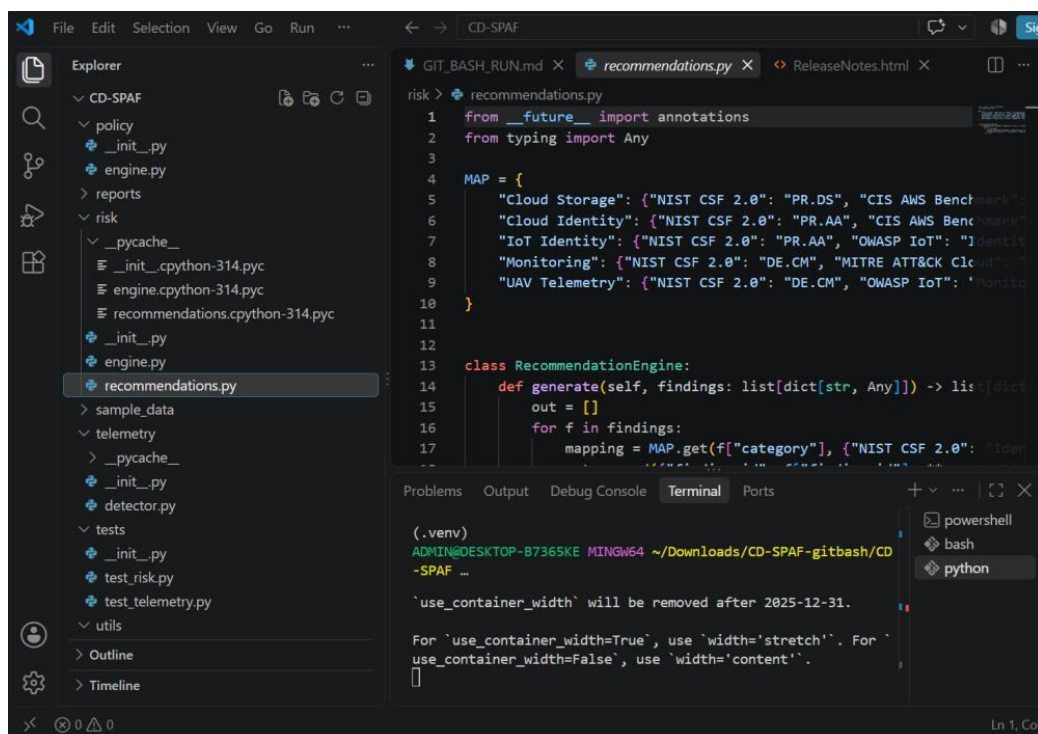


Figure 5.4: Risk and Security Recommendation Mapping

Figure 5.4 shows the recommendation stage that follows. A mapping table in recommendations.py associates each finding category with a NIST CSF 2.0 category and, where relevant, a further source of guidance: Cloud Storage findings are mapped to PR.DS (Data Security) and the CIS AWS Benchmark; Cloud Identity findings to PR.AA (Identity Management, Authentication and Access Control) and the CIS AWS Benchmark; IoT Identity findings to PR.AA and OWASP IoT guidance; Monitoring findings to DE.CM (Continuous Monitoring) and MITRE ATT&CK Cloud; and UAV Telemetry findings to DE.CM and OWASP IoT

guidance [6], [30], [35], [36]. The purpose of this mapping is to avoid producing a finding with no potential remediation direction, so that an identified issue can be connected to a recognised security practice rather than left as a bare observation.

5.6 Dashboard Implementation

The security assessment dashboard is the prototype's main user-facing component, implemented with Streamlit, a Python framework for interactive data applications that provides functions for presenting metrics, charts and tables through a web interface (Figure 5.5).

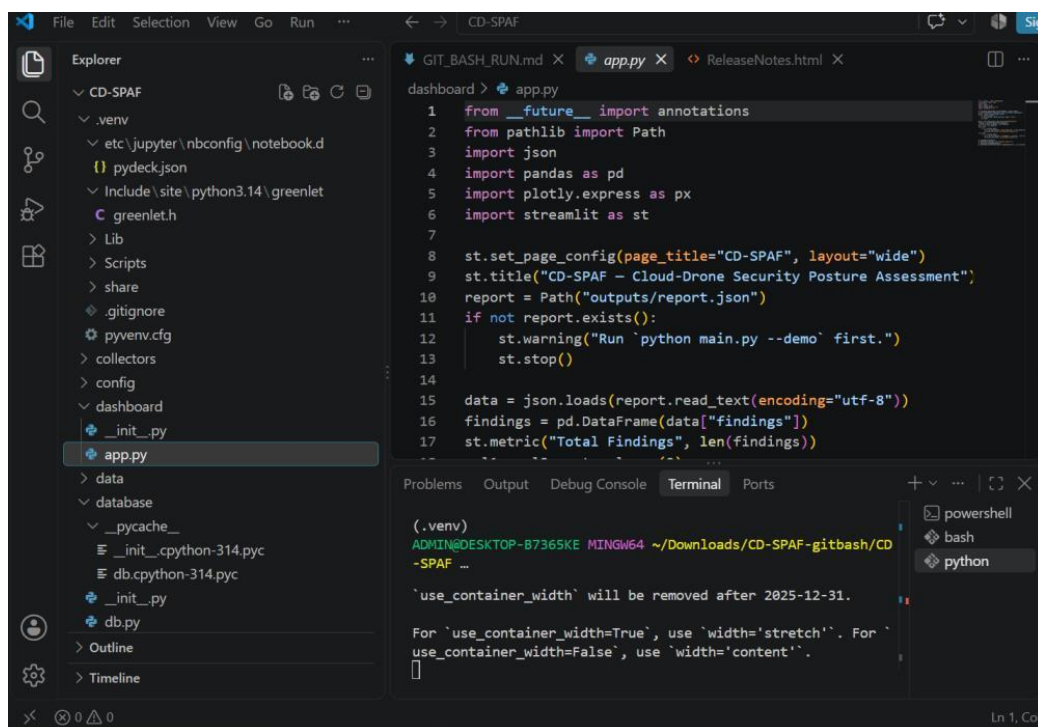


Figure 5.5: Streamlit Dashboard Implementation

The dashboard reads the generated assessment report from outputs/report.json and processes the findings it contains to present key information to the user, including a Total Findings metric calculated directly from the assessment output rather than entered manually, so that the displayed count always reflects the contents of the current report. It also presents graphical summaries — a severity distribution and risk scores by category, shown in Figure 5.8 — alongside tabular findings and recommendations. The dashboard design was kept deliberately simple: the intention was not to build a complex enterprise security-operations interface, but to demonstrate how the output of the CD-SPAF assessment engine could be presented in a form a researcher or practitioner can review directly.

5.7 Integration of the CD-SPAF Components

The individual modules described above were integrated into the complete CD-SPAF prototype. The workflow begins with collecting or loading security information, which can include cloud configuration data, device identity information, system information and UAV telemetry. The telemetry component processes UAV-related information and identifies unusual observations while the policy engine, in parallel, evaluates relevant security information against the defined policies; both produce findings that the risk engine evaluates according to the model above, after which the recommendation engine maps the finding's category to relevant guidance and stores the result in the assessment report. The dashboard then reads this report and presents it visually. The complete flow —

*Collect Security Information → Analyse Telemetry → Evaluate Policies →
Generate Findings → Calculate Risk → Map Recommendations → Display
Dashboard*

— corresponds directly to the workflow presented in Figure 4.3. This modular integration also supports future development: a more advanced cloud scanner, for example, could be introduced without changing the telemetry detector, and a more advanced anomaly-detection model could be added without redesigning the dashboard.

5.8 Prototype Execution and Assessment Output

The integrated prototype was executed to demonstrate the operation of CD-SPAF as a whole. The assessment run generated 36 security findings, which represent conditions identified by the implemented assessment components and should not be interpreted as 36 confirmed, exploitable vulnerabilities. The dashboard groups this output by severity and assessment category, producing findings across UAV telemetry, cloud storage, cloud identity, IoT/device identity and monitoring, which demonstrates that the framework can combine information from different security domains into one assessment result — directly relevant to the research problem, since UAV operations connected to cloud environments cross multiple security boundaries. A security issue affecting a cloud storage component, for example, may have a different technical cause from an unusual UAV telemetry observation, yet both can be represented within the same assessment framework and assigned risk information on a common basis.

	finding_id	title	severity	asset_id	category	description
1	TEL-ISO-53	Telemetry anomaly detected (IsolationForest)	High	UAV-001	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.
2	TEL-ISO-68	Telemetry anomaly detected (IsolationForest)	High	UAV-001	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.
3	TEL-ISO-77	Telemetry anomaly detected (IsolationForest)	High	UAV-001	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.
4	TEL-ISO-98	Telemetry anomaly detected (IsolationForest)	High	UAV-001	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.
5	TEL-ISO-275	Telemetry anomaly detected (IsolationForest)	High	UAV-001	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.
6	TEL-ISO-355	Telemetry anomaly detected (IsolationForest)	High	UAV-001	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.
7	TEL-ISO-375	Telemetry anomaly detected (IsolationForest)	High	UAV-001	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.
8	TEL-ISO-411	Telemetry anomaly detected (IsolationForest)	High	UAV-001	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.
9	TEL-ISO-413	Telemetry anomaly detected (IsolationForest)	High	UAV-001	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.
10	TEL-DBS-0	Telemetry anomaly detected (DBSCAN)	High	UAV-001	UAV Telemetry	DBSCAN flagged an unusual telemetry observation.

Figure 5.6: CD-SPAF Security Findings Dashboard

The findings output also provides supporting evidence explaining why a particular condition was identified, including the relevant security data, the affected category, the applicable rule and the resulting severity (Figure 5.7). A posture assessment should give a reviewer enough information to understand the basis of a finding, since a numerical score with no supporting evidence provides limited practical value.

	category	description	evidence	score
0	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.	["timestamp":"2026-01-01 00:00:19","speed":13.115463537868463,"battery":97.13195]	64.6
1	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.	["timestamp":"2026-01-01 00:00:53","speed":16.40123426110605,"battery":94.012128]	64.6
2	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.	["timestamp":"2026-01-01 00:01:08","speed":14.60122672246776,"battery":92.796896]	64.6
3	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.	["timestamp":"2026-01-01 00:01:17","speed":17.056579550213105,"battery":91.92455]	64.6
4	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.	["timestamp":"2026-01-01 00:01:38","speed":13.752594320419362,"battery":89.82956]	64.6
5	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.	["timestamp":"2026-01-01 00:04:35","speed":12.0420395690062,"battery":69.808145]	64.6
6	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.	["timestamp":"2026-01-01 00:05:55","speed":12.83911979195516,"battery":60.594976]	64.6
7	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.	["timestamp":"2026-01-01 00:06:15","speed":12.656162080259872,"battery":57.93597]	64.6
8	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.	["timestamp":"2026-01-01 00:06:51","speed":16.01351731453719,"battery":54.903327]	64.6
9	UAV Telemetry	IsolationForest flagged an unusual telemetry observation.	["timestamp":"2026-01-01 00:06:53","speed":14.26968718241337,"battery":53.312095]	64.6

Figure 5.7: CD-SPAF Security Findings and Evidence

Figure 5.8 shows the dashboard's severity distribution and risk-score-by-category output for this run: of the 36 findings, the large majority were classified as High severity and a smaller proportion as Medium severity. This classification supports prioritisation within the prototype rather than representing a universal, industry-defined severity calculation, and the risk scores

shown should be interpreted within the assumptions set out in Chapter 3. A detailed statistical interpretation of these results, including the exact severity percentages and category breakdown, is provided in Chapter 6.

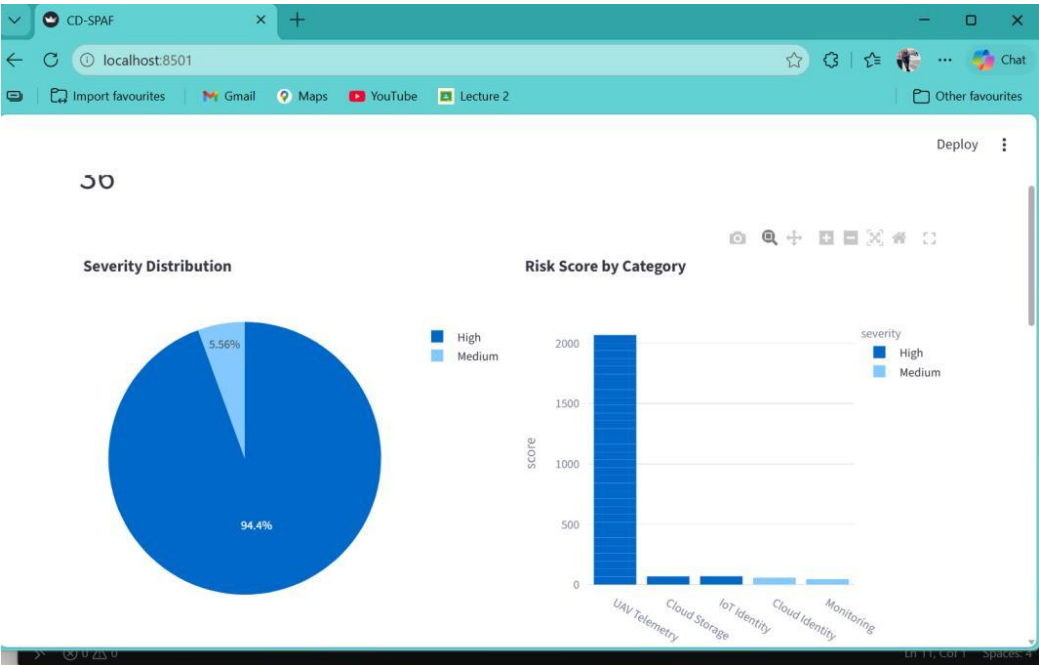


Figure 5.8: CD-SPAF Dashboard — Severity Distribution and Risk Score by Category

Figure 5.9 shows an extract of the underlying findings table, in this case DBSCAN-detected telemetry observations together with their evidence and computed score; this table is the source of the totals discussed in Chapter 6.

The figure shows a screenshot of the CD-SPAF dashboard with the 'Findings' table expanded. The table contains 10 rows of data, each representing a DBSCAN-detected telemetry observation. The columns are: id, category, description, evidence, and score. The evidence column contains JSON objects with timestamp, speed, and battery information. The score column shows a consistent value of 64.6 for all entries.

id	category	description	evidence	score
19	UAV Telemetry	DBSCAN flagged an unusual telemetry observation.	{"timestamp":"2026-01-01 00:00:09","speed":12.68191668622511,"battery":99.060954}	64.6
20	UAV Telemetry	DBSCAN flagged an unusual telemetry observation.	{"timestamp":"2026-01-01 00:00:10","speed":9.777336443559339,"battery":98.92496}	64.6
21	UAV Telemetry	DBSCAN flagged an unusual telemetry observation.	{"timestamp":"2026-01-01 00:00:11","speed":14.596148624135239,"battery":99.0068}	64.6
22	UAV Telemetry	DBSCAN flagged an unusual telemetry observation.	{"timestamp":"2026-01-01 00:00:12","speed":11.09865946123755,"battery":99.03559}	64.6
23	UAV Telemetry	DBSCAN flagged an unusual telemetry observation.	{"timestamp":"2026-01-01 00:00:13","speed":9.02082838226667,"battery":97.768147}	64.6
24	UAV Telemetry	DBSCAN flagged an unusual telemetry observation.	{"timestamp":"2026-01-01 00:00:14","speed":10.118921322744656,"battery":97.7839}	64.6
25	UAV Telemetry	DBSCAN flagged an unusual telemetry observation.	{"timestamp":"2026-01-01 00:00:15","speed":10.162118710779021,"battery":98.4308}	64.6
26	UAV Telemetry	DBSCAN flagged an unusual telemetry observation.	{"timestamp":"2026-01-01 00:00:16","speed":12.09391197358356,"battery":97.95052}	64.6
27	UAV Telemetry	DBSCAN flagged an unusual telemetry observation.	{"timestamp":"2026-01-01 00:00:17","speed":11.507069090016772,"battery":98.6510}	64.6
28	UAV Telemetry	DBSCAN flagged an unusual telemetry observation.	{"timestamp":"2026-01-01 00:00:18","speed":11.393702458246933,"battery":97.8899}	64.6
29	UAV Telemetry	DBSCAN flagged an unusual telemetry observation.	{"timestamp":"2026-01-01 00:00:19","speed":13.115463537868463,"battery":97.1319}	64.6

Figure 5.9: CD-SPAF Findings Table Extract — DBSCAN-Detected Telemetry Observations

5.9 Security Recommendations Output

The final output of the prototype is the generation of security recommendations, produced according to the category of each finding and the guidance mapped within the recommendation engine, creating a chain from finding, to category, to security framework or benchmark, to recommended action. Figure 5.10 shows two extracts of the recommendations table. In the upper extract, telemetry-related findings are mapped to monitoring guidance under NIST CSF 2.0 (DE.CM), while the lower extract shows the final rows of the table, in which the AWS-specific and device findings — covering S3 storage, IAM permissions, device certificates and centralised logging — are mapped to the CIS AWS Benchmark and MITRE ATT&CK Cloud references where available. This provides an additional layer of usefulness beyond a simple vulnerability or configuration scanner: the objective is not only to identify an issue but to provide a structured direction for improving the security posture, although any recommendation generated from a general mapping may still need adaptation to the operational requirements of a specific UAV system, cloud architecture and organisational policy.

The screenshot displays the CD-SPAF web interface. The top table, titled 'Recommendations', lists findings 0 through 10. The bottom table, also titled 'Recommendations', lists findings 26 through 35. Both tables include columns for finding_id, NIST CSF 2.0, OWASP IoT, Remediation, CIS AWS Benchmark, and MITRE ATT&CK Cloud.

	finding_id	NIST CSF 2.0	OWASP IoT	Remediation	CIS AWS Benchmark	MITRE ATT&CK Cloud
0	TEL-ISO-19	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
1	TEL-ISO-53	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
2	TEL-ISO-68	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
3	TEL-ISO-77	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
4	TEL-ISO-98	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
5	TEL-ISO-275	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
6	TEL-ISO-355	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
7	TEL-ISO-375	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
8	TEL-ISO-411	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
9	TEL-ISO-413	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
10	TEL-DBS-16	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
26	TEL-DBS-16	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
27	TEL-DBS-17	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
28	TEL-DBS-18	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
29	TEL-DBS-19	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
30	TEL-THR-120	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
31	TEL-THR-275	DE.CM	Monitoring	Investigate anomalous telemetry, validate mission constraints, and improve telemetr	None	None
32	AWS-S3-001	PR.DS	None	Enable encryption at rest and block unintended public access.	S3 data protection	None
33	AWS-IAM-001	PR.AA	None	Remove wildcard permissions and enforce least privilege.	IAM least privilege	None
34	UAV-CERT-001	PR.AA	Identity and	Use a unique certificate/identity per drone and rotate credentials.	None	None
35	AWS-LOG-001	DE.CM	None	Enable centralised audit logging and appropriate retention/alerting.	None	Cloud logging and mon

Figure 5.10: CD-SPAF Security Recommendations Output (Extract)

5.10 Implementation Security and Safety Considerations

The implementation was designed around a controlled, authorised assessment environment, and no destructive testing or unauthorised access to third-party systems was required for the prototype. This was particularly important given the cloud-connected UAV context: unrestricted security testing against real UAV infrastructure or third-party cloud systems could create operational, legal and safety risks that fall outside the scope of an MSc research project. The implementation instead focuses on assessing controlled security information and UAV telemetry, demonstrating the assessment process without requiring interference with a live UAV mission or an external system, and keeps security assessment clearly separated from active exploitation — CD-SPAF identifies conditions and produces findings, but does not attempt to exploit any weakness it detects, consistent with the ethical and methodological boundary established in Chapter 3.

5.11 Implementation Limitations

Several limitations apply to the implementation itself. It represents a research prototype rather than a production security platform, demonstrating the framework using controlled inputs rather than providing enterprise-level cloud security monitoring. The UAV component is based on telemetry data rather than a continuously operating physical fleet, so it demonstrates telemetry analysis without reproducing every condition that could arise during real-world operations. The machine-learning component identifies statistical anomalies but cannot independently determine whether an anomaly is malicious, so additional human or contextual analysis would be needed before treating an anomaly as a confirmed incident. The cloud assessment component is limited to the security information available within the prototype and should not be read as a replacement for the native security services provided by AWS or Microsoft Azure, and the risk-scoring approach is specific to this research and would require further validation — using larger datasets, real operational environments and comparison with established risk-assessment methodologies — before broader claims could be made about its effectiveness. These limitations are considered further in the evaluation and discussion chapters that follow.

5.12 Chapter Summary

This chapter presented the implementation of CD-SPAF as a modular Python prototype comprising a main execution component, a UAV telemetry anomaly detector, a security policy engine, a risk assessment component, a recommendation engine, a report generation mechanism and a Streamlit dashboard. The telemetry module applies Isolation Forest and DBSCAN to identify unusual observations, the policy engine evaluates YAML-defined security rules, and both feed findings into the risk assessment process, where they receive risk scores and severity classifications under the research-specific model before the recommendation engine maps them to NIST CSF 2.0, CIS, OWASP-related and MITRE ATT&CK-related guidance. The dashboard then presents this output visually. The prototype generated 36 security findings during the demonstrated run, providing the evidence base for the results and analysis presented in Chapter 6.

CHAPTER 6 — RESULTS AND ANALYSIS

This chapter presents and analyses the results obtained from the CD-SPAF prototype described in Chapter 5. The purpose of the evaluation was to determine whether the implemented system could process UAV-related telemetry and cloud security information, identify security conditions, assign risk information, and generate security recommendations, using the controlled prototype environment described in Chapter 3. The results should therefore be interpreted within the scope of the implemented system rather than as evidence of performance across real-world UAV and multi-cloud environments, and the analysis does not treat the generated findings as confirmed exploitable vulnerabilities; they represent security conditions identified by the implemented assessment rules and telemetry analysis that require attention or further investigation.

6.1 Overview of Assessment Results

The CD-SPAF prototype generated 36 security findings during the demonstrated assessment run, as shown earlier in the dashboard output (Figures 5.6 and 5.9). This total provides an initial indication of the amount of security information identified during the assessment, but on its own it does not indicate the significance of the results, which is why CD-SPAF additionally presents findings by severity and by category rather than relying on the overall count alone. The findings span multiple security areas rather than concentrating on a single component, which is consistent with CD-SPAF's objective of assessing the security posture of cloud-connected UAV operations across both UAV-related and cloud-related domains.

6.2 Security Finding Categories

Table 6.1 summarises the five categories present in the assessment output.

Table 6.1: CD-SPAF Security Finding Categories

Finding Category	Security Area
UAV Telemetry	UAV operational and telemetry behaviour
Cloud Storage	Security of cloud-based storage resources
Cloud Identity	Identity and access-related security
IoT Identity	Device / UAV identity and authentication
Monitoring	Security monitoring and visibility

The UAV Telemetry category, produced by the anomaly-detection component described in section 5.3, dominates the findings table. The recommendations extracts in Figure 5.10 show that findings 0 to 31 in the assessment output are telemetry findings — ten with the TEL-ISO prefix (Isolation Forest), twenty with the TEL-DBS prefix (DBSCAN) and two with the TEL-THR prefix (threshold check) — while the remaining four findings each correspond to one of the other categories: a cloud storage condition (AWS-S3-001), a cloud identity condition (AWS-IAM-001), a device identity condition (UAV-CERT-001) and a monitoring condition (AWS-LOG-001). Table 6.2 summarises this observed distribution.

Table 6.2: Observed Distribution of Findings by Category

Category	Findings	Approx. Share
UAV Telemetry	32	88.9%
Cloud Storage	1	2.8%
Cloud Identity	1	2.8%
IoT Identity	1	2.8%
Monitoring	1	2.8%
Total	36	100%

This distribution is a feature of the specific controlled scenario used to demonstrate the prototype rather than a general statement about the relative frequency of these condition types in real UAV-cloud deployments: the telemetry dataset used for this run was deliberately constructed to contain a larger number of anomalous readings, which is why telemetry findings dominate the total. The four non-telemetry findings likewise correspond one-to-one to the four controlled posture conditions defined in the policy file, so their number reflects the design of the scenario rather than a discovery rate. Even so, the presence of all five categories in a single output confirms that the framework is not restricted to one type of security condition, and that UAV-related and cloud-related findings can be represented within the same assessment result — the property most directly relevant to the research question.

6.3 Severity Distribution

The CD-SPAF dashboard indicates that, of the 36 findings, 94.4% were classified as High severity and 5.56% as Medium severity, corresponding to 34 High-severity findings and 2 Medium-severity findings, as shown in Figure 5.8 and Table 6.3. No findings were classified as Low or Critical in this run. The colour coding of the risk-score chart in Figure 5.8 indicates that the two Medium findings are the Cloud Identity and Monitoring findings, while all 32 telemetry findings and the Cloud Storage and IoT Identity findings were classified as High.

Table 6.3: Severity Distribution of Generated Findings

Severity	Percentage	Number of Findings
High	94.4%	34
Medium	5.56%	2
Total	100%	36

This result should be interpreted carefully. A high proportion of High-severity findings does not automatically mean that the tested environment contains a large number of confirmed critical vulnerabilities; the classification is an output of the CD-SPAF prototype and reflects the assumptions and scoring rules defined for this research, and in particular the composition of the demonstration dataset described above, which was weighted towards anomalous telemetry. The result nevertheless demonstrates that the framework can distinguish between findings according to assessed importance rather than producing an unstructured list of observations, and the presence of a smaller Medium-severity group confirms that the assessment logic is capable of producing more than one severity outcome.

6.4 Risk Score Analysis

In addition to severity classification, the dashboard provides risk scores for each assessed category, shown in the right-hand panel of Figure 5.8. UAV Telemetry received a markedly higher aggregate risk score than the other four categories, consistent with both its higher finding count and the High-severity classification assigned to most telemetry-related findings, while Cloud Storage, Cloud Identity, IoT Identity and Monitoring each returned comparatively low scores, consistent with each contributing a single finding to the demonstrated run. Because the underlying formula multiplies asset criticality, exposure, control failure and mission impact, a category with more findings and a higher average severity will necessarily

accumulate a higher aggregate score under this model; the result confirms that the scoring mechanism behaves consistently with its design rather than establishing an independent, external judgement of which security domain is objectively riskier in general. The findings table also shows that every telemetry finding received the same individual risk score of 64.6 (Figures 5.7 and 5.9), so the aggregate for the category — approximately 2,067 across 32 findings — is in effect the number of telemetry findings multiplied by a constant. Under the current parameterisation, therefore, the model does not distinguish between a mild and a pronounced telemetry anomaly; the factor values applied to telemetry findings do not vary with the size of the deviation.

6.5 Relationship Between Findings and Recommendations

Every finding produced by the assessment run was mapped to a recommendation by the recommendation engine described in section 5.5. The telemetry-related findings were consistently mapped to NIST CSF 2.0's Detect (DE.CM) guidance together with a monitoring-oriented remediation instruction — investigate the anomalous telemetry, validate mission constraints and improve telemetry monitoring — while the four non-telemetry findings each received a category-specific mapping: the storage finding to S3 data-protection guidance under the CIS AWS Benchmark, the identity finding to IAM least-privilege guidance, the device-identity finding to certificate-rotation guidance, and the monitoring finding to centralised logging guidance with a MITRE ATT&CK Cloud reference. This confirms that the mapping logic responds to the category of a finding rather than applying a single generic recommendation to every output, which was one of the specific design requirements set out in section 4.6.

6.6 Assessment Consistency

Because the assessment operates on a fixed, controlled dataset and a defined set of policy rules, repeated execution of the prototype against the same input data was observed to produce the same findings, severity classifications and risk scores. This observation was informal: no statistical repeatability testing was carried out, and because Isolation Forest is a randomised algorithm, identical output across runs depends on its random state being fixed in the implementation. Within those limits, the behaviour supports the methodological

objective, set out in Chapter 3, of producing an assessment process whose behaviour can be examined and reproduced rather than one that varies unpredictably between runs.

6.7 Chapter Summary

The demonstrated assessment run produced 36 security findings distributed across five categories, dominated by UAV telemetry findings but including one finding each for cloud storage, cloud identity, device identity and monitoring. Severity classification placed 94.4% of findings (34) as High and 5.56% (2, the Cloud Identity and Monitoring findings) as Medium, and the risk-scoring mechanism produced aggregate category scores consistent with each category's finding count and severity. Every finding was successfully mapped to a category-appropriate recommendation, and repeated execution against the same controlled input produced consistent results. These results confirm that the prototype can execute the complete CD-SPAF workflow described in Chapter 4 end to end; what they mean for the framework's underlying research contribution is considered in the next chapter.

CHAPTER 7 — DISCUSSION

Chapter 6 reported what the CD-SPAF prototype produced during its demonstrated assessment run. This chapter considers what those results mean in relation to the literature reviewed in Chapter 2, the research objectives and research question set out in Chapter 1, and the framework's practical value as a defensive security tool.

7.1 Interpretation of the Assessment Findings

The headline result — 36 findings, 94.4% High severity, spread across five categories — is best read as a demonstration of mechanism rather than a measurement of real-world UAV-cloud risk. The high proportion of High-severity findings follows directly from the composition of the controlled telemetry dataset, which was deliberately weighted towards anomalous readings so that the anomaly-detection and risk-scoring components would have sufficient material to exercise. It would therefore be inaccurate to present the 94.4% figure as evidence that cloud-connected UAV systems are typically found to be in a high-risk state; the figure instead shows that, when presented with a dataset containing a large number of anomalous observations, CD-SPAF's scoring model consistently classifies them as high priority rather than diluting them across the severity scale. This distinction between what an assessment tool reports and what that report implies about the wider population of possible deployments is one that any user of CD-SPAF — and any reviewer of this dissertation — needs to keep in view.

The more informative result, from a research perspective, is that the four non-telemetry findings — covering cloud storage, cloud identity, device identity and monitoring — were generated, classified and mapped to recommendations through exactly the same pipeline as the 32 telemetry findings. This is the direct evidence that the framework's central design goal, a shared assessment pathway for UAV-side and cloud-side conditions, was actually achieved in the implementation rather than remaining only a design intention. Because those four conditions were predefined in the policy file, the result shows that the pipeline can carry cloud-side findings through to scored recommendations; it does not show that the scanner can discover such conditions in a live account.

7.2 Relationship Between UAV and Cloud Security

Chapter 2 argued that UAV security research and cloud security practice have developed largely in parallel, and that this separation makes it difficult to see the combined security condition of a cloud-connected UAV deployment. The results in Chapter 6 offer a small-scale but concrete counter-example: a single assessment report contained findings from both domains, each carrying a severity, a risk score and a recommendation expressed in the same structure. This does not resolve the underlying technical differences between, say, an S3 bucket permission and a telemetry anomaly — those remain different kinds of problem requiring different remediation skills — but it does show that the two kinds of finding can be represented, compared and prioritised alongside one another without forcing an analyst to consult two separate tools and reconcile their outputs manually. This is the practical form that the research contribution described in section 1.7 takes in the working prototype.

7.3 Telemetry Anomaly Detection in Context

The telemetry component's reliance on Isolation Forest and DBSCAN situates CD-SPAF within a well-established line of work on statistical and density-based anomaly detection, rather than proposing a new detection algorithm of its own [38], [39]. The literature on GNSS spoofing and UAV communication attacks reviewed in section 2.2 shows that a determined attacker can produce telemetry that looks superficially plausible to a simple threshold check, which is one reason the anomaly-based approach used here — flagging statistical deviation rather than checking against fixed limits — is preferable to a naive implementation, even within a research prototype. At the same time, the results in Chapter 6 illustrate the corresponding limitation directly: because every one of the 32 telemetry findings was drawn from a dataset engineered to contain anomalies, the demonstration says relatively little about the detector's false-positive or false-negative behaviour against realistic operational telemetry, where genuine anomalies are typically rare events against a background of normal variation. This gap between demonstrating that a mechanism functions and establishing that it is accurate is characteristic of prototype-stage security research and is addressed further in section 7.9 and in the future work identified in Chapter 8.

7.4 Policy-Based Assessment and Risk Scoring in Context

The policy engine's YAML-based rules and the multiplicative risk model reflect two ideas drawn from the literature: policy-as-code as a mechanism for expressing security

requirements independently of a specific cloud provider [24], [34], and structured, factor-based risk scoring as recommended in general terms by NIST SP 800-30 [33]. The results in section 6.4 show the risk model behaving exactly as its formula predicts — categories with more findings and higher average severity accumulate higher aggregate scores — which confirms that the mechanism is implemented correctly but does not, on its own, validate the formula's four weighting factors (asset criticality, exposure, control failure and mission impact) as an accurate representation of real UAV-cloud risk. That validation would require comparing the model's output against expert risk judgements or historical incident data, neither of which was available within the scope of this project. The identical score of 64.6 assigned to every telemetry finding (section 6.4) illustrates the point: the mechanism is consistent, but its parameters do not yet reflect differences in the severity of individual anomalies. The contribution at this stage is a working, internally consistent scoring mechanism that later research can calibrate and test, rather than a validated risk metric in its own right.

7.5 Multi-Cloud Considerations

The prototype implementation processes a single controlled dataset rather than live parallel AWS and Azure environments, so the multi-cloud claim made in Chapter 1 needs to be read at the level of design rather than of demonstrated operation. The architecture in Chapter 4, and specifically the separation between policy rules and provider-specific collectors described in section 2.13, is structured so that AWS-specific and Azure-specific checks could be added as additional collectors without changing the policy engine, the risk engine or the dashboard. The recommendation mapping shown in section 5.5, which already distinguishes AWS-specific guidance such as the CIS AWS Benchmark from more general NIST CSF references, is consistent with this design intention. What the current implementation demonstrates is that the architecture does not hard-code a single provider's assumptions into the core assessment logic; it does not yet demonstrate simultaneous, live assessment of a real multi-cloud deployment, which remains future work.

7.6 Achievement of Research Objectives

Revisiting the five objectives set out in section 1.4: the first objective, identifying security weaknesses in cloud-supported UAV architectures, was achieved through the literature

review in Chapter 2 and reflected in the five finding categories implemented in the prototype. The second, designing CD-SPAF around NIST CSF 2.0 and Zero Trust principles, was achieved at the design level described in Chapter 4, with the caveat noted throughout that these principles were used as informing references rather than fully implemented standards. The third objective, developing a prototype capable of collecting cloud configuration information and UAV or controlled telemetry and applying predefined assessment rules, was demonstrated by the working implementation in Chapter 5 with one qualification: cloud configuration information was represented by controlled posture conditions in the policy file rather than collected from live accounts, so the collection step was demonstrated for telemetry but only simulated for cloud configuration. The fourth, implementing risk assessment that categorises findings by severity, was demonstrated by the severity and risk-score output analysed in sections 6.3 and 6.4. The fifth objective, evaluating the prototype using controlled scenarios and determining whether it generates useful recommendations, was addressed in section 6.5, where every finding received a category-appropriate mapping. Overall, the objectives were met at the level appropriate to a controlled MSc research prototype; none required claiming production-scale validation, live multi-cloud deployment or confirmed vulnerability discovery, none of which were undertaken.

7.7 Answer to the Research Question

The research question asked how a unified security framework can assess and reduce cloud, IoT and UAV-related risks in drone systems that depend on AWS or Azure for telemetry, storage, command services, analytics and monitoring. The evidence developed in this dissertation suggests a specific, bounded answer: by separating information collection from policy evaluation and risk scoring, as CD-SPAF does, a single framework can represent security-relevant conditions from UAV telemetry and from cloud configuration in a common structure — finding, category, evidence, severity, risk score and recommendation — allowing them to be reviewed and prioritised together rather than through separate, provider- or domain-specific tools. The prototype demonstrates that this is achievable at a functional level. It does not demonstrate, and this dissertation does not claim, that the approach measurably reduces risk in an operational UAV fleet; that would require deployment and evaluation beyond the scope available here, and is identified explicitly as future work in Chapter 8.

7.8 Practical Significance and Contribution

For a practitioner managing a cloud-connected UAV deployment, the practical value demonstrated here is consolidation: rather than checking AWS or Azure security dashboards, a UAV telemetry monitoring tool and a separate compliance spreadsheet independently, a CD-SPAF-style assessment could in principle surface all of these as a single prioritised list with consistent severity language and category-appropriate remediation guidance. This is a modest but genuine contribution given the current fragmentation described in section 2.15 — it does not require an organisation to replace any existing security control, only to add an integration and reporting layer above them. The contribution is best framed, consistent with section 1.7, as a feasibility demonstration and an architectural pattern rather than as a finished product: the specific value of the research lies in showing that the integration is technically achievable with a modest, modular Python implementation, not in the particular policies, risk weights or recommendation mappings encoded in the current prototype, which would need considerable further development before operational use.

7.9 Limitations and Implications

The limitations identified in sections 3.12 and 5.11 have direct implications for how these results should be used. Because the telemetry dataset was engineered rather than operational, the severity and risk-score results cannot be generalised to real UAV fleets, and any organisation considering a similar framework would need to re-validate the anomaly-detection thresholds and risk weights against its own historical or live data before relying on the output for decision-making. Because the cloud assessment component checks a limited, illustrative set of conditions rather than the full range of AWS and Azure security services, CD-SPAF should be understood as a complementary layer that sits above native cloud security tooling, not a replacement for it — a point already made in section 2.8 but worth restating here as an implication for use rather than only a limitation of design. Finally, because the risk-scoring formula has not been validated against expert judgement or incident data, its numerical output should be treated as a relative prioritisation device within a single assessment run rather than as a calibrated measure that can be compared meaningfully across different organisations or time periods. None of these limitations undermine the central finding that a unified UAV-cloud assessment pipeline is technically feasible; they

define the conditions under which that feasibility result would need to be extended before the framework could support real operational decisions.

CHAPTER 8 — CONCLUSION AND FUTURE WORK

8.1 Conclusion

This dissertation set out to design, implement and evaluate CD-SPAF, a defensive Cloud-Drone Security Posture Assessment Framework for UAV operations that depend on cloud infrastructure across AWS and Azure. The literature review in Chapter 2 established that UAV cybersecurity research and cloud security practice have largely developed as separate fields, and that existing tools — CSPM platforms, Zero Trust guidance, NIST CSF 2.0 and provider-specific security services — each address part of the cloud-connected UAV security problem without combining UAV-side and cloud-side findings into a single assessment. CD-SPAF was designed and implemented to address this gap by analysing UAV telemetry with anomaly-detection techniques, evaluating controlled cloud and device posture conditions against YAML policy rules, assigning risk scores through a shared model, and generating category-appropriate recommendations, all presented through a single Streamlit dashboard.

The controlled evaluation in Chapters 6 and 7 confirmed that this pipeline operates end to end: the demonstrated assessment run produced 36 findings across five categories — UAV Telemetry, Cloud Storage, Cloud Identity, IoT Identity and Monitoring — with 94.4% classified as High severity and 5.56% as Medium, and every finding was successfully mapped to a category-appropriate recommendation drawn from NIST CSF 2.0, the CIS AWS Benchmark, OWASP IoT guidance or MITRE ATT&CK Cloud. These results demonstrate technical feasibility rather than measured real-world risk reduction, a distinction maintained throughout this dissertation and one that any future use of the framework should preserve.

8.2 Research Contribution

The primary contribution is the design and working prototype implementation of CD-SPAF itself: a modular architecture and Python implementation that integrates cloud security posture assessment, UAV telemetry anomaly detection, policy-as-code evaluation, risk scoring and recommendation mapping within one framework, evidenced by the complete implementation described in Chapter 5 and the end-to-end results in Chapter 6. A secondary contribution is the comparative analysis in Chapter 2, which positions CD-SPAF explicitly against existing UAV security research, cloud security guidance and posture-assessment

approaches, clarifying that its value lies in integration rather than in a novel detection technique or a novel cloud security control. The framework and its component architecture also provide a concrete foundation that future research can extend, calibrate and validate, rather than a finished operational product.

8.3 Research Limitations

The main limitations of this research, discussed in more detail in sections 3.12, 5.11 and 7.9, are summarised here for completeness. The prototype was evaluated within a controlled research environment using simulated or controlled UAV telemetry rather than a live physical UAV fleet, and the demonstration dataset was constructed to contain a high proportion of anomalous readings, which limits the extent to which the severity and risk-score results can be generalised. The cloud assessment component covers a limited, illustrative set of conditions rather than the full range of AWS and Azure security services, cloud posture conditions were predefined rather than collected from live accounts, and no live, simultaneous multi-cloud deployment was assessed. The risk-scoring formula, while internally consistent, has not been validated against expert judgement, historical incident data or an established industry benchmark. Finally, the scope of assessment scenarios reflects the resources available to an MSc project rather than the range of conditions that a production security tool would need to cover.

8.4 Future Work

Several directions follow directly from the limitations above.

1. Full AWS and Azure integration: extending the cloud security scanner to connect to live AWS and Azure accounts through their respective APIs, rather than processing prepared configuration data, and validating the provider-agnostic policy design described in section 2.13 against real, simultaneous multi-cloud environments.
2. Real-time UAV telemetry: replacing the controlled CSV-based dataset with a live or near-real-time telemetry feed, for example from PX4 or ArduPilot SITL, MAVLink or an IoT ingestion service, to evaluate the anomaly detector against operational data containing a realistic, low base rate of genuine anomalies.

3. Physical and multi-UAV testing: extending evaluation to a physical UAV, and subsequently to a small fleet, once the necessary safety, ethical and organisational approvals are in place.
4. Improved anomaly detection: evaluating additional or hybrid machine-learning techniques, and calibrating detection thresholds against labelled operational data to establish false-positive and false-negative rates that the current demonstration does not provide.
5. A larger security policy library: expanding the YAML-based rule set to cover a wider range of AWS and Azure services, IoT device-management conditions, and additional NIST CSF and CIS control areas.
6. Continuous security monitoring: moving from a single assessment run to scheduled or continuous re-assessment, consistent with the Zero Trust principle of ongoing rather than one-off verification discussed in section 2.9.
7. Validation of the risk model: testing the four-factor risk-scoring formula against expert security assessments or historical incident data to determine whether its weightings and multiplicative structure produce risk rankings consistent with real-world outcomes.
8. SIEM and SOC integration: exporting CD-SPAF findings to a security information and event management platform so that they can be correlated with other organisational security data rather than reviewed only through the standalone dashboard.
9. Automated incident response: investigating whether high-confidence, high-severity findings could trigger predefined, human-approved remediation actions, building on the recommendation mappings already implemented in the current prototype.

8.5 Researcher Reflection

Looking back on the project, the most significant lesson concerned scope. The original proposal envisaged read-only collectors for both AWS and Azure, a simulated UAV environment and a unified dashboard. As the design developed, it became clear that each of these elements was a substantial piece of work in its own right, and that attempting all of

them at production depth would have left little time for evaluation. Narrowing the prototype to controlled posture conditions and a CSV telemetry dataset was, in hindsight, the right decision for an MSc project, but it also meant accepting that the cloud side of CD-SPAF would be demonstrated at the level of design and data structure rather than live integration. Writing accurately about that boundary proved almost as demanding as the implementation itself.

Designing CD-SPAF also changed my understanding of what integration means in security assessment. Initially I treated it largely as a technical problem of combining outputs. In practice, the harder question was semantic: a telemetry anomaly and a storage misconfiguration are different kinds of evidence, and representing them in one finding structure required decisions about categories, evidence fields and severity that are not neutral. The observation that every telemetry finding received an identical risk score was an instructive example. The pipeline worked as designed, yet the result showed how much the usefulness of a risk model depends on the parameters chosen for it.

Working with Isolation Forest and DBSCAN reinforced the difference between detecting a statistical outlier and identifying a security event. Without labelled data it was not possible to say how many of the flagged observations were meaningful, and I came to regard the absence of accuracy metrics as a limitation to be stated clearly rather than worked around. Mapping findings to NIST CSF 2.0, the CIS AWS Benchmark, OWASP IoT guidance and MITRE ATT&CK Cloud showed, similarly, that these sources operate at very different levels of abstraction, which makes a simple category-level mapping convenient but inevitably approximate.

If the research were repeated, I would build a small labelled telemetry dataset early, so that the anomaly detectors could be evaluated rather than only demonstrated, and I would implement at least one live read-only cloud collector against a dedicated test account before extending the policy library. I would also ask practitioners to review the risk factors before relying on them. Overall, the project developed my understanding of UAV and cloud security as a single problem space, and of the discipline needed to separate what a prototype demonstrates from what it merely suggests.

REFERENCES

- [1] H. Shakhathreh et al., "Unmanned aerial vehicles (UAVs): A survey on civil applications and key research challenges," *IEEE Access*, vol. 7, pp. 48572–48634, 2019, doi: 10.1109/ACCESS.2019.2909530.
- [2] D. He, S. Chan, and M. Guizani, "Communication security of unmanned aerial vehicles," *IEEE Wireless Commun.*, vol. 24, no. 4, pp. 134–139, Aug. 2017, doi: 10.1109/MWC.2016.1600073WC.
- [3] R. Altawy and A. M. Youssef, "Security, privacy, and safety aspects of civilian drones: A survey," *ACM Trans. Cyber-Phys. Syst.*, vol. 1, no. 2, pp. 1–25, 2017, doi: 10.1145/3001836.
- [4] J.-P. Yaacoub, H. Noura, O. Salman, and A. Chehab, "Security analysis of drones systems: Attacks, limitations, and recommendations," *Internet Things*, vol. 11, Art. no. 100218, Sep. 2020, doi: 10.1016/j.iot.2020.100218.
- [5] Z. Wang et al., "A survey on cybersecurity attacks and defenses for unmanned aerial systems," *J. Syst. Archit.*, vol. 138, Art. no. 102870, May 2023, doi: 10.1016/j.sysarc.2023.102870.
- [6] National Institute of Standards and Technology, "The NIST Cybersecurity Framework (CSF) 2.0," NIST, Gaithersburg, MD, USA, NIST CSWP 29, Feb. 2024, doi: 10.6028/NIST.CSWP.29.
- [7] S. Rose, O. Borchert, S. Mitchell, and S. Connelly, "Zero trust architecture," NIST, Gaithersburg, MD, USA, NIST SP 800-207, Aug. 2020, doi: 10.6028/NIST.SP.800-207.
- [8] Y. Mekdad, A. Aris, L. Babun, A. El Fergougui, M. Conti, R. Lazzeretti, and A. S. Uluagac, "A survey on security and privacy issues of UAVs," *Comput. Netw.*, vol. 224, Art. no. 109626, Apr. 2023, doi: 10.1016/j.comnet.2023.109626.
- [9] N. Bai, X. Hu, and S. Wang, "A survey on unmanned aerial systems cybersecurity," *J. Syst. Archit.*, vol. 156, Art. no. 103282, Nov. 2024, doi: 10.1016/j.sysarc.2024.103282.
- [10] B. Cordill, D. Fang, and S. Xu, "A comprehensive survey of security and privacy in UAV systems," *IEEE Access*, vol. 13, pp. 117843–117866, 2025, doi: 10.1109/ACCESS.2025.3583985.
- [11] Y. Zeng, R. Zhang, and T. J. Lim, "Wireless communications with unmanned aerial vehicles: Opportunities and challenges," *IEEE Commun. Mag.*, vol. 54, no. 5, pp. 36–42, May 2016, doi: 10.1109/MCOM.2016.7470933.
- [12] Y. Zeng, J. Lyu, and R. Zhang, "Cellular-connected UAV: Potential, challenges, and promising technologies," *IEEE Wireless Commun.*, vol. 26, no. 1, pp. 120–127, Feb. 2019, doi: 10.1109/MWC.2018.1800023.
- [13] M. Asadpour, B. Van den Bergh, D. Giustiniano, K. A. Hummel, S. Pollin, and B. Plattner, "Micro aerial vehicle networks: An experimental analysis of challenges and opportunities," *IEEE Commun. Mag.*, vol. 52, no. 7, pp. 141–149, Jul. 2014, doi: 10.1109/MCOM.2014.6852096.
- [14] M. L. Psiaki and T. E. Humphreys, "GNSS spoofing and detection," *Proc. IEEE*, vol. 104, no. 6, pp. 1258–1270, Jun. 2016, doi: 10.1109/JPROC.2016.2526658.

- [15] A. Altaweel, H. Mukkath, and I. Kamel, "GPS spoofing attacks in FANETs: A systematic literature review," *IEEE Access*, vol. 11, pp. 55233–55280, 2023, doi: 10.1109/ACCESS.2023.3281731.
- [16] A. A. Eshmawi, M. Umer, I. Ashraf, and Y. Park, "Enhanced machine learning ensemble approach for securing small unmanned aerial vehicles from GPS spoofing attacks," *IEEE Access*, vol. 12, pp. 27344–27355, 2024, doi: 10.1109/ACCESS.2024.3359700.
- [17] A. Finn, M. Jia, Y. Li, and J. Yuan, "Detecting stealthy GPS spoofing attack against UAVs using onboard sensors," in *Proc. IEEE INFOCOM Workshops (INFOCOM WKSHPS)*, May 2024, pp. 1–6, doi: 10.1109/INFOCOMWKSHPS61880.2024.10620818.
- [18] M. Ceccato, F. Formaggio, and S. Tomasin, "Spatial GNSS spoofing against drone swarms with multiple antennas and Wiener filter," *IEEE Trans. Signal Process.*, vol. 68, pp. 5782–5794, 2020, doi: 10.1109/TSP.2020.3028752.
- [19] J. Ding, C. Tang, L. Zhang, Z. Yue, Y. Liu, and Z. Dan, "UAV communication and navigation signals jamming methods," in *Proc. IEEE Int. Conf. Signal Process., Commun. Comput. (ICSPCC)*, Aug. 2024, pp. 1–6, doi: 10.1109/ICSPCC62635.2024.10770465.
- [20] K.-Y. Tsao, T. Girdler, and V. G. Vassilakis, "A survey of cyber security threats and solutions for UAV communications and flying ad-hoc networks," *Ad Hoc Netw.*, vol. 133, Art. no. 102894, Aug. 2022, doi: 10.1016/j.adhoc.2022.102894.
- [21] İ. Bekmezci, O. K. Sahingoz, and Ş. Temel, "Flying ad-hoc networks (FANETs): A survey," *Ad Hoc Netw.*, vol. 11, no. 3, pp. 1254–1270, May 2013, doi: 10.1016/j.adhoc.2012.12.004.
- [22] R. Roman, J. Lopez, and M. Mambo, "Mobile edge computing, Fog et al.: A survey and analysis of security threats and challenges," *Future Gener. Comput. Syst.*, vol. 78, pp. 680–698, Jan. 2018, doi: 10.1016/j.future.2016.11.009.
- [23] H. Sedjelmaci, S. M. Senouci, and N. Ansari, "Intrusion detection and ejection framework against lethal attacks in UAV-aided networks: A Bayesian game-theoretic methodology," *IEEE Trans. Intell. Transp. Syst.*, vol. 18, no. 5, pp. 1143–1153, May 2017, doi: 10.1109/TITS.2016.2600370.
- [24] Cloud Security Alliance, *Security Guidance for Critical Areas of Focus in Cloud Computing v4.0*. Seattle, WA, USA: CSA, 2017. [Online]. Available: <https://cloudsecurityalliance.org/artifacts/security-guidance-v4>
- [25] Amazon Web Services, "Security pillar – AWS Well-Architected Framework," AWS Documentation. Accessed: Sep. 25, 2026. [Online]. Available: <https://docs.aws.amazon.com/wellarchitected/latest/security-pillar/welcome.html>
- [26] Microsoft, "Microsoft cloud security benchmark v1 overview," Microsoft Learn. Accessed: Sep. 25, 2026. [Online]. Available: <https://learn.microsoft.com/en-us/security/benchmark/azure/overview-mcsb-v1>
- [27] Microsoft, "Microsoft Defender for Cloud documentation," Microsoft Learn. Accessed: Sep. 25, 2026. [Online]. Available: <https://learn.microsoft.com/en-us/azure/defender-for-cloud/>

- [28] Amazon Web Services, "Security in AWS IoT," AWS IoT Core Developer Guide. Accessed: Sep. 25, 2026. [Online]. Available: <https://docs.aws.amazon.com/iot/latest/developerguide/security.html>
- [29] Microsoft, "Secure your IoT solutions," Microsoft Learn (Azure IoT). Accessed: Sep. 25, 2026. [Online]. Available: <https://learn.microsoft.com/en-us/azure/iot/iot-overview-security>
- [30] OWASP Foundation, "OWASP Internet of Things Top 10," 2018. Accessed: Sep. 25, 2026. [Online]. Available: <https://owasp.org/www-project-internet-of-things/>
- [31] A. R. A. Rahumath Ali and P. Deepalakshmi, "Cloud security posture management: Automating risk detection, compliance enforcement, and vulnerability remediation in cloud infrastructure," in *Proc. 1st Int. Conf. Sustainability Technol. Adv. Eng. Domain (SUSTAINED)*, Dec. 2024, pp. 1–6, doi: 10.1109/SUSTAINED63638.2024.11073903.
- [32] R. Chandramouli and Z. Butcher, "A zero trust architecture model for access control in cloud-native applications in multi-location environments," NIST, Gaithersburg, MD, USA, NIST SP 800-207A, Sep. 2023, doi: 10.6028/NIST.SP.800-207A.
- [33] Joint Task Force Transformation Initiative, "Guide for conducting risk assessments," NIST, Gaithersburg, MD, USA, NIST SP 800-30 Rev. 1, Sep. 2012, doi: 10.6028/NIST.SP.800-30r1.
- [34] Open Policy Agent, "Open Policy Agent documentation." Accessed: Sep. 25, 2026. [Online]. Available: <https://www.openpolicyagent.org/docs/>
- [35] Center for Internet Security, "CIS Amazon Web Services Foundations Benchmark." Accessed: Sep. 25, 2026. [Online]. Available: https://www.cisecurity.org/benchmark/amazon_web_services
- [36] The MITRE Corporation, "Enterprise cloud matrix," MITRE ATT&CK. Accessed: Sep. 25, 2026. [Online]. Available: <https://attack.mitre.org/matrices/enterprise/cloud/>
- [37] F. Pedregosa et al., "Scikit-learn: Machine learning in Python," *J. Mach. Learn. Res.*, vol. 12, no. 85, pp. 2825–2830, 2011.
- [38] F. T. Liu, K. M. Ting, and Z.-H. Zhou, "Isolation forest," in *Proc. 8th IEEE Int. Conf. Data Mining (ICDM)*, Dec. 2008, pp. 413–422, doi: 10.1109/ICDM.2008.17.
- [39] M. Ester, H.-P. Kriegel, J. Sander, and X. Xu, "A density-based algorithm for discovering clusters in large spatial databases with noise," in *Proc. 2nd Int. Conf. Knowl. Discovery Data Mining (KDD-96)*, 1996, pp. 226–231.