Detection and Identification of Covert and Replay Attacks in Cyber-physical Systems Using Model-Based and Data-Driven Based Methods

Kimia Firoozi

A Thesis

in

The Department

of

Electrical and Computer Engineering

Presented in Partial Fulfillment of the Requirements

for the Degree of

Master of Applied Science (Electrical and Computer Engineering) at

Concordia University

Montréal, Québec, Canada

December 2024

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Kimia Firoozi**

Entitled: **Detection and Identification of Covert and Replay Attacks in Cyber-physical Systems Using Model-Based and Data-Driven Based Methods**

and submitted in partial fulfillment of the requirements for the degree of

Master of Applied Science (Electrical and Computer Engineering)

complies with the regulations of this University and meets the accepted standards with respect to originality and quality.

Signed by the Final Examining Committee:

_____ Chair
Dr. M. Reza Soleymani

_____ External Examiner
Dr. Javad Dargahi

_____ Examiner
Dr. M. Reza Soleymani

_____ Supervisor
Dr. Khashayar Khorasani

Approved by _____
Dr. Yousef R. Shayan, Chair
Department of Electrical and Computer Engineering

_____ 2024

_____
Dr.Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Detection and Identification of Covert and Replay Attacks in Cyber-physical Systems Using Model-Based and Data-Driven Based Methods

Kimia Firoozi

Cyber-physical systems (CPSs) have revolutionized various domains, including smart grids, manufacturing, transportation systems, and autonomous vehicles such as Unmanned Aerial Vehicles (UAVs). While CPS configurations offer numerous advantages, they are particularly vulnerable to stealthy cyber-attacks due to their specific structure, in which the command and control center is far from the plant and operation side, making security a critical concern. Among these attacks, covert and replay attacks pose significant challenges for detection, particularly in UAV applications. This research focuses on detecting such stealthy attacks using two approaches: a model-based scenario where the mathematical model of the system is available and a data-driven scenario where only data is accessible, making it more practical for high-fidelity systems.

In the model-based approach, the research develops a coding design to enhance the detection functionality and expand its scope to meet security requirements. This framework strengthens the system's ability to counter stealthy attacks effectively. On the other hand, for scenarios where only data is available and inherently vulnerable to cyber-attacks, an effective algorithm is designed to detect and identify covert and replay attacks by assigning appropriate labels. This algorithm leverages neural network (NN) models for training and evaluation, ensuring high accuracy and proficiency in attack detection.

To optimize computational efficiency, the algorithm employs feature selection techniques during the data preprocessing stage, minimizing the reliance on complex NN models. This not only reduces computational resource consumption but also enhances the accuracy of the detection model.

The effectiveness of the proposed methodologies is validated through simulations on a 6-degree-of-freedom quadrotor, a critical application highly susceptible to cyber-attacks. The results demonstrate the efficiency and reliability of the contributions in detecting and mitigating stealthy cyber-attacks in CPS configurations.

This research provides a framework for improving the security of CPSs in both model-based and data-driven scenarios, contributing to safer and more resilient systems in critical applications.

To my soulmate

Dad

Acknowledgments

First, I express my deepest gratitude to my supervisor, **Prof. Khashayar Khorasani**, for his invaluable support, patience, and motivation throughout this journey. His belief in my abilities and the life-changing opportunity to work under his guidance have been transformative. Trusting me with this opportunity marked a pivotal moment in my academic and professional growth.

I would also like to extend my heartfelt thanks to my family, especially brothers, **Milad and Soroush**, for their unwavering support after the passing of my late father. Words cannot fully capture my gratitude for their constant encouragement, sacrifices, and unconditional love. I appreciate my mother, **Prof. Shahla Rostami**, whose strength and resilience have been a guiding light, teaching me to stay strong and never give up in the face of challenges. She has been my greatest inspiration, and I am forever grateful to her.

I am profoundly thankful to my partner **Mohamad** and my cute puppy **Bentley** for their endless support, positivity, and encouragement throughout this journey. Their unwavering love and belief in me have been a source of strength and comfort during the most demanding times.

I extend my sincere appreciation to **Dr. Ali Rezaeifar and his family** for their invaluable support in helping me secure this position and for guiding me through the fundamental steps of this professional journey.

Finally, I sincerely thank my colleagues and friends at Concordia University, **Mehdi, Rezvan, Sina, and Shahab**, for their support and encouragement. Special thanks to **Mohamadreza Nematollahi**, whose extensive knowledge and generous, moment-by-moment assistance were invaluable. I am also immensely grateful to **Elham** for her continuous kindness and support. Together, their selfless help played a crucial role in the success of this journey, and I will always cherish their contributions.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

The recent development of Cyber-Physical Systems (CPS) has been recognized as a significant achievement since the interaction between cyber and physical elements is growing in various domains. This kind of system is a complex combination of computational and physical processes. Embedded systems, sensors, and actuators interact in real time to coordinate different operations in different domains. The progress of CPS is supported by notable advancements in embedded systems, sensor and actuator technologies, and the growth of the Internet of Things (IoT), which have fundamentally changed how digital and physical elements are integrated.

This powerful system goes beyond the fundamental interaction of physical and computational elements; it combines these components to produce more spectacular systems that are more remarkable than the essential combination of their parts. CPS is utilized across various domains, including healthcare, transportation, environmental monitoring, and the military. A cyber-physical system is greatly enhanced by incorporating new communication technologies. More than any other area, the rapidly increasing sub-domain of unmanned aerial vehicles (UAVs) has made researchers more interested in this category. CPS strengthens, and its capabilities solidify its status as a crucial element of modern technological infrastructure. Figure 1.1 illustrates the setup of this system within a limited scope and perfectly shows the absence of humans in the communication channel. Data transmission occurs entirely without human presence, allowing human operators only to control or monitor the primary system operations remotely and to issue commands to achieve the desired objective.

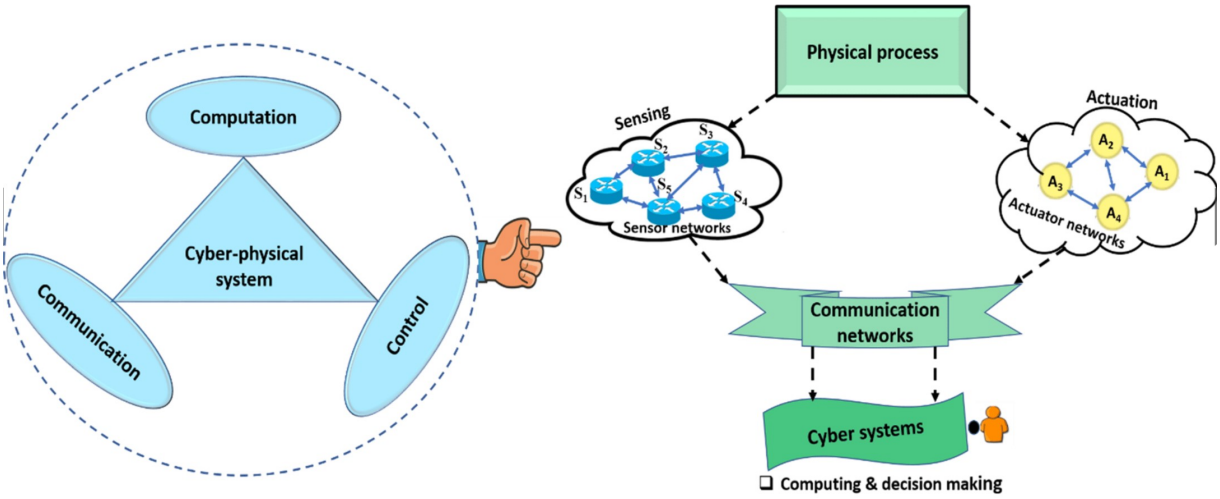

Figure 1.1: System includes sensors and actuators communicating with central processing [1].

However, this integration brings new challenges, particularly in security aspects. This kind of system's security involves not only safeguarding the physical processes that can interact with the control system but also protecting the data and communication networks in the unique nature of CPS, where cyber vulnerabilities can control physical consequences, requires revising traditional security methods.

Securing CPS, especially in the context of UAVs such as quadrotors, becomes vital. These systems, increasingly being utilized for various purposes, are vulnerable to sophisticated cyber-attacks that could compromise their data integrity, availability, confidentiality, and physical functionality. The possibility of these attacks being hidden and carried out by individuals with extensive knowledge of the system adds to the difficulty, making detecting and preventing such threats problematic.

Regarding the dynamic nature of today's environment, ensuring robust security measures in CPSs, specifically in connection to UAVs, is one of the highest priority. The field is currently at a critical point, necessitating novel strategies that effectively tackle existing security concerns and proactively anticipate potential weaknesses in the future. The thesis aims to enhance the current understanding of the relationship between cyber and physical components related to security issues. It also investigates how developing technologies might be utilized to strengthen these systems against sophisticated attacks.

1.1 Problem Statement

This thesis aims to address securing CPS which they are vulnerable to sophisticated covert and replay cyber-attacks. These attacks are particularly challenging as they are executed by adversaries with comprehensive knowledge of the system, enabling them to manipulate actuators and sensors discreetly. Such covert attacks can mask the effects of injected harmful inputs, making the data appear normal to the Command and Control (C&C) systems and rendering the attacks undetectable to common defence mechanisms. This vulnerability poses a significant threat to the integrity of unmanned systems.Also, a replay attack is able to record the data for a certain period of time and then pose the manipulated reference and make the system vulnerable; however, in the meantime, the recorded data is sent to the monitoring centre periodically, which makes this attack undetectable as well.

Figure 1.2: UAV as a perspective of cyber-physical system [2].

The research focuses on developing innovative detection strategies in model-based and data-driven approaches. The initial phase involves creating a detection coding matrix to identify covert or replay attacks separately, which improves the chi-square method within the C&C framework.

Subsequently, the study explores the application of neural networks in the data-driven detection methodology. This approach improves on traditional supervised machine learning by using advanced methods, such as random forests to highlight important features and a straightforward neural network model to train with the selected feature. This combination allows for both the detection and differentiation of covert and replay attacks. Unlike methods that only detect attacks, this approach also identifies the type of attack, which is a valuable step forward in strengthening the security of cyber-physical systems. These methods are expected to enhance the accuracy and speed of detection compared to other existing methods.

The thesis investigates two distinct scenarios. The first scenario involves detecting covert or replay attacks using a state-space representation system, resolved through a coding-filtering methodology within this framework. The second scenario addresses situations where only a limited dataset from the UAV is available, and the complete system model is inaccessible due to security constraints. This scenario focuses on detecting covert or replay attack with respect to the available data and come up with label of each attack also emphasizes the need for rapid and accurate detection methods, considering the limited data and absence of an entire system model.

This research aims to contribute to unmanned system security significantly as evaluated case study, offering robust and efficient solutions to detect these attacks in CPS configuration, consequently ensuring these systems' safe and reliable operation in specific missions.

1.2 Literature Review

When designing the research plan, it is necessary to prioritize developing a clear and focused path that smoothly guides through applicable and essential information from previous researches. This systematic approach prevents deviation from the primary goal and focuses on valuables that significantly establish the study target. The research wants to methodically build this path to extract crucial information from previous studies. This will establish a strong foundation for the investigation and allow for a focused exploration of relevant areas that are key to the study's objectives. Emphasizing the importance of a well-defined path is crucial for making significant contributions to research goals.

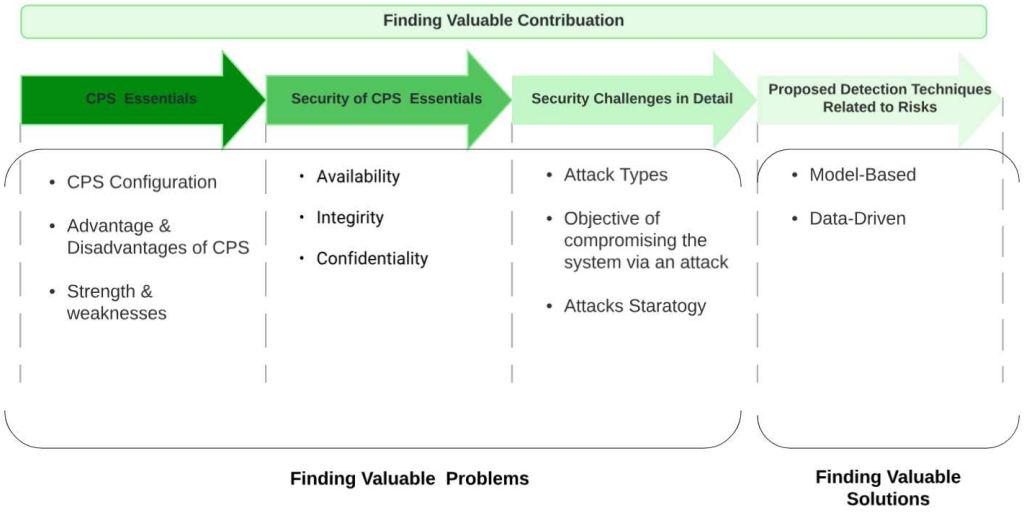

Figure 1.3: Structure for literature review.

The illustration in Figure1.3 highlights the essential pathway that requires credit for finding the most influential idea on this subject. This methodology not only helps to find valuable information to define concrete problems but also enables a thorough comparison with similar work in this field, assuring consistency and solid solutions.

1.2.1 Chronology of Cyber-Physical Systems

A cyber-physical system background presents a multifaceted view, underscoring the critical integration of computation with physical processes. Lee in [12] identifies significant design challenges, especially concerning physical components' unique safety and reliability requirements, distinct from those in general-purpose computing. The paper argues persuasively that existing computing and networking technologies are foundations for CPS, necessitating a complete update of these abstractions to integrate physical dynamics with computation effectively. These challenges question the sufficiency of current technologies. Its emphasis lies in the necessity of aligning computational and networking abstractions with physical dynamics, and this paper offers concrete methods in this regard. However, the security aspect is still the main concern of this paper. [13] Discussion by characterizing CPS and summarizing research from varied perspectives, including energy control

and secure control, thus underscoring the diversity and applicability of CPS. These studies excel in providing a multi-domain viewpoint and in highlighting potential applications. Still, they fail to address specific technical challenges and propose detailed solutions for the identified research gaps [14], [15]. Several years later, with the advancement of machine learning as a potent representation of CPS, data collection has become a significant concern in this domain. The author introduces a new data-collection prototype system, targeting real-time capabilities and hybrid-system qualities. While this approach is practical, its scope is relatively narrow, focusing primarily on data collection and not addressing broader design and theoretical challenges in CPS [16]. These articles provide valuable insights into the complexities and potential applications of CPS. Yet, they consistently emphasize the need for more concrete solutions and innovative CPS design and implementation approaches. Additionally, the vital issue concerning the security of such systems, considering their unique configuration, continues to be a critical challenge for these powerful systems.

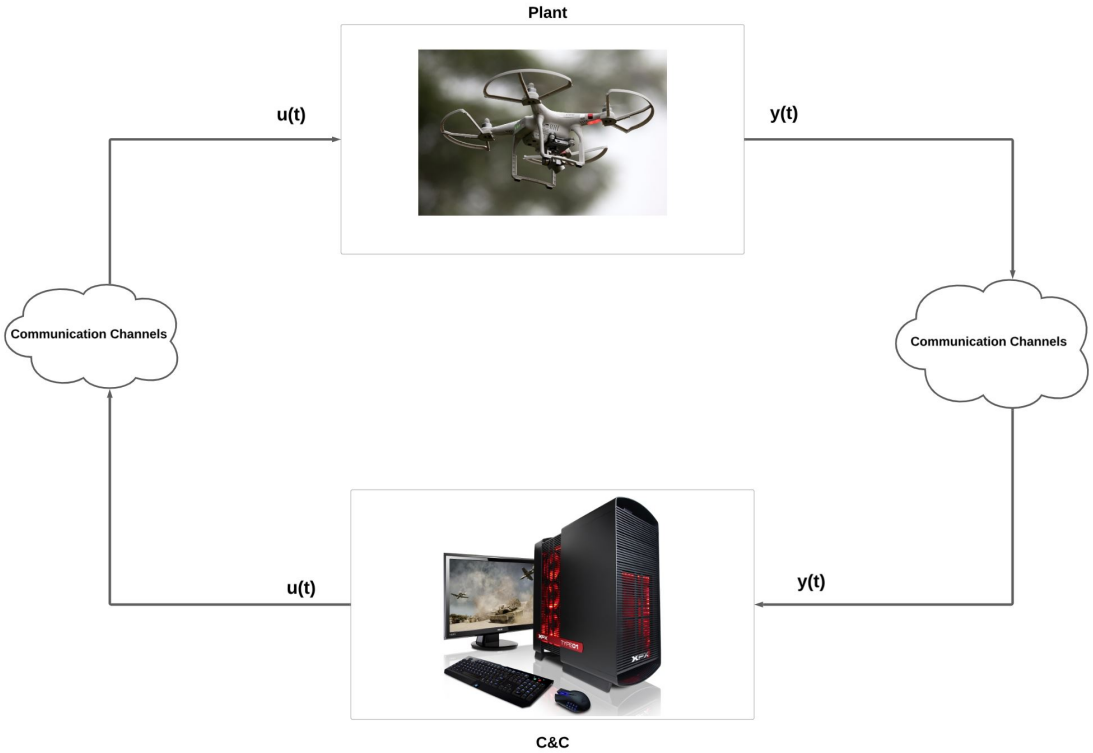

Figure 1.4: Cyber-physical systems configuration.

Figure 1.4 provides a detailed schematic representation of these systems, illustrating that the primary operations occur within the plant while all monitoring and command processes are in the control center. These two components are linked via communication channels.

1.2.2 CPS Security and their Challenges

Ensuring the security of CPS is the highest priority in this field and requires a solid foundation of knowledge. Cyber-security has traditionally focused on two different aspects. The first perspective of view has three fundamental features: availability, integrity, and confidentiality (CIA) [17]. Confidentiality ensures that unauthorized parties cannot access sensitive data, while integrity guarantees that data remains unaltered by unauthorized entities. Availability is essential for timely access to data and system functionalities. For CPS, the security landscape is unique, as cyber-attacks can impact both the computational and physical aspects due to feedback loops. This is one of the critical categorizations that all the cyber threats in the CPS domain could compromise or disrupt one or more of these three features. As the field of CPS security continues to evolve, research efforts, guided by foundational references with significant citations, are laser-focused on developing strategies to safeguard the confidentiality, integrity, and availability of critical data and system functionalities. The second approach investigated in the literature review is the categories of attack properties resulting in different attack types. These dimensions are disclosure resources, disruption resources, and system knowledge. A cyber-attack could occur if one, two, or all of these parameters could be accessible for the attacker [3].

The categorization presented in Figure 1.5 offers a comparative perspective on vulnerabilities resources, describing the viewpoints of both attackers and defenders within a single illustration. From the attacker's perspective, the focus is on exploiting system knowledge, disrupting resources, and exposing confidential information (disclosure resources). Conversely, from the defender's standpoint, the attacker's actions compromise the system's availability, confidentiality, and integrity, undermining the system's security framework.

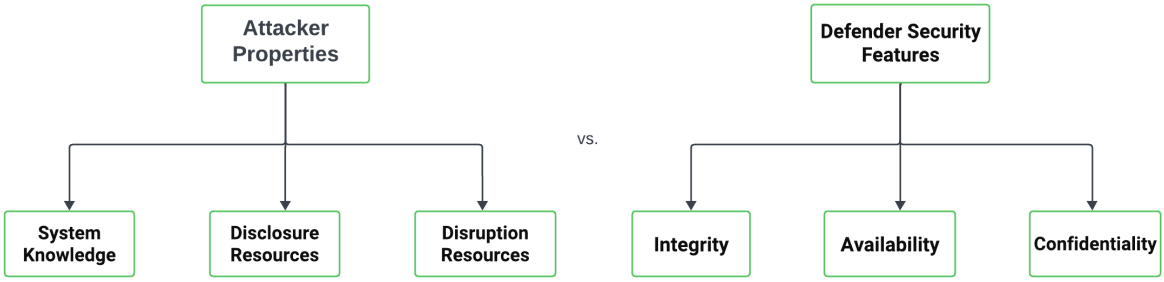

Figure 1.5: Security features from defender vs. attack properties from attacker.

By examining Figure 1.5 can realize the concept from both perspectives which is the attacker sides view aiming to launch an attack, and the defender, striving to maintain the system's security.

UAV's Security and their Challenges: The security challenges in cyber-physical systems, particularly in the sub-domain of UAVs, have been examined in several research papers, each offering valuable insights. However, these papers have been found to vary in terms of comprehensiveness. In [18], the emerging trend of attackers employing commercial off-the-shelf small unmanned aerial systems (UASs) against malicious activities was highlighted, and countermeasures were proposed, although they lack in-depth technical details. Conversely, [19] investigated UAV vulnerabilities, presented a specific cyber threat, and discussed potential countermeasures. However, its focus remained primarily on a singular vulnerability. In [20] acknowledged UAV vulnerability to various attacks and emphasized the significance of robust security protocols. Nevertheless, it primarily analyzed existing protocols and vulnerabilities without proposing concrete solutions. [21] addressed compromised power electronics in UAV flight computers, proposing the potential security solution of a machine learning-based intrusion detection system. Despite these contributions, there remains a need for a more comprehensive examination that encompasses specific vulnerabilities, provides concrete solutions, and offers a complete view of security challenges in CPS with UAVs.

The fundamental security features are crucial for understanding CPS's security challenges. They are essential for formulating efficient strategies in the following phases of this study. Understanding the relationship between cyber and physical risks is crucial. It promotes the creation of inventive methods to connect them. Implementing these strategies is vital for improving the resilience and strength of CPS against cyber threats. Moreover, a comprehensive understanding of different attack

types and their categorizations is crucial for developing a customized detection approach for specific attacks.

1.2.3 Attack Categories in Cyber-Physical Systems

Attack classifications in CPSs are critical for choosing the proper solution for the detection and identification of each attack while dealing with the wide range of threats that may compromise the security and reliability of these interconnected systems. The attack taxonomy provides an organized framework for categorizing and analyzing attacker activity, providing significant insights for developing effective solutions. In this part of the literature review, the existing body of information around attack classifications in CPS is analyzed to illuminate the landscape of cyber threats. Various attacks were investigated in [3], including denial-of-service (DoS) attacks, bias injection attacks, replay attacks, and covert attacks, focusing on the sophisticated nature of covert attacks.

Figure 1.6 illustrates the various attack strategies, as previously discussed, indicating that each attack utilizes specific properties to execute the attack. It details different kinds of attacks and outlines the characteristics required to carry out a cyber-attack, rendering the system vulnerable. Also, there exists the relation and connection between these three attacks as well, as noted in [22], how covert attacks, which aim to remain undetected, can be launched in various forms, including through Denial-of-Service (DoS) attacks, to disrupt the integrity and functionality of physical processes. The authors emphasize that covert DoS attacks target communication channels and control signals to impair the system without immediate detection.

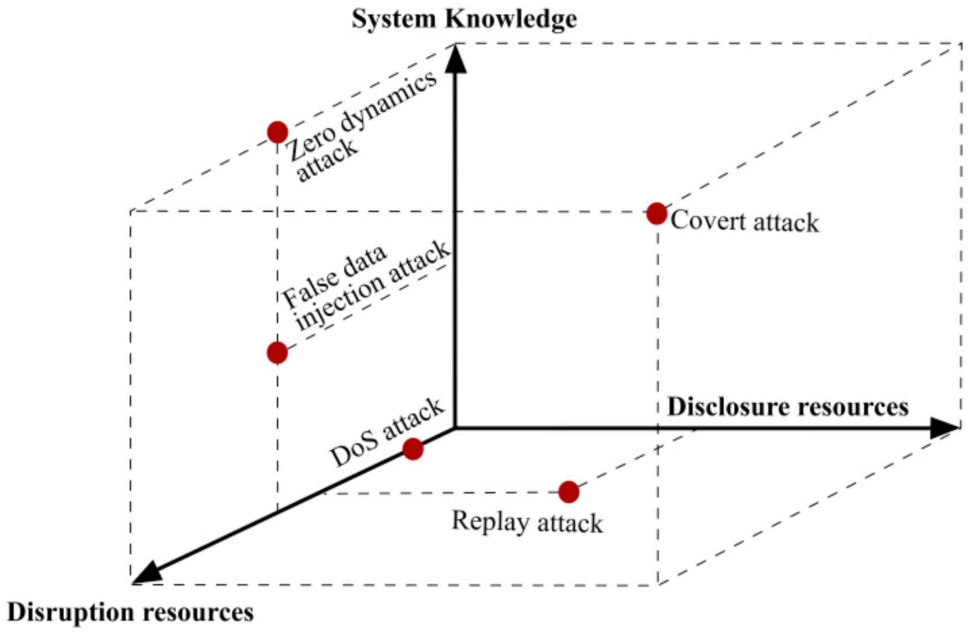

Figure 1.6: Attack types and their properties [3].

As shown in Figure 1.6, attacks, characterized by bombarding systems with excessive requests, cause severe service disruptions. While numerous studies underscore the vulnerability of CPS to such attacks like [23], highlighting their potential to cause significant outages, a notable weakness in existing research is the limited exploration of effective mitigation strategies for these systems. This gap underscores the need for further investigation into robust defence mechanisms against attacks. An increasing emphasis has been observed on characterizing and mitigating attacks across various CPS contexts. The research [24] is noted for its exploration into DoS attack patterns, proposing novel defence mechanisms crafted explicitly for the unique challenges of CPS infrastructures. However, these studies often exhibit a limitation in their scope, focusing narrowly and potentially overlooking broader CPS scenarios. On the other hand, the study by [25] clarifies the evolving nature of DoS attack strategies. It underscores the necessity for adaptive security measures, highlighting a critical trend towards dynamic defence mechanisms responding to changing cyber threats. This progression in the field, along with identified research gaps, forms a core part of the current literature landscape as a subset of covert attacks.

Bias injection attacks modify data within a CPS to add bias, which results in inaccurate judgments or actions. This type of attack is especially dangerous in healthcare systems and self-driving cars. Researchers presented detection techniques to detect and neutralize bias injection attacks in CPS [26].

Replay attacks are identified as a dangerous cyber threat in the literature review. These attacks pose a significant risk to the reliability and trustworthiness of sensor data and control commands, possibly compromising the integrity of CPS operations. This issue has led researchers to explore a range of mitigation techniques. A significant focus has been placed on cryptographic and timestamp-based methods to enhance the resilience of CPS against replay attacks. However, the literature also indicates a limitation in these studies, as they may not fully address the diverse and evolving nature of such attacks [27]. Several significant references help to understand replay attack mitigation in CPS. [28] provided an in-depth review of mitigation strategies, highlighting current trends and issues in the field while providing insights into the strengths and limitations of cryptographic and timestamp-based methods [29] extended their investigation by conducting a comparison analysis of the encryption algorithms used for replay attack avoidance in CPS, rating their effectiveness and computing overhead to help experts and researchers choose the best solution. Furthermore, [30] improved the detection and mitigation of replay attacks in CPS by emphasizing the role of formal methods and the importance of temporal logic in ensuring system security. However, [31] discusses a moving target defence(MTD) strategy for replay attack detection by altering system control signals in a randomized manner. Their approach includes periodic adjustments to control signals, which prevents attackers from effectively mirroring past control commands to carry out a replay attack. By making system behaviour unpredictable, this method enables the detection of abnormal behaviour, making replay attacks detectable and less effective. Moreover, [32] gathered the survey highlighting their role in enhancing security in CPSs by continuously altering system configurations to thwart attacks. The study summarized that MTD's unpredictability disrupts static attacks like replay, as attackers cannot rely on consistent data patterns.

Covert attacks are the most sophisticated and challenging, as shown in Figure 1.6; adversaries use extensive information about the targeted system, including its actuators and sensor measurement data, and have complete knowledge of the system. Attackers operate the system in a way that

covers their malicious operations, effectively making the system's output undetectable from regular functioning. This form of danger has received special attention in the context of UAV-based CPS, emphasizing its serious consequences [33]. This attack takes different forms, each posing distinct difficulties to CPS security and integrity. The basic covert attack is one such type, in which control signals are quietly modified to decrease system performance while avoiding detection [34]. This strategy allows attackers to control the system's state trajectory under certain situations. [35] focused on the complex nature of covert attacks driven by data, providing a valuable understanding of the vital function of Markov parameters in this particular context that innovates new undetectable covert attacks and the defence strategy concerning this specific type of covert attack.

The field of covert attacks has expanded further with undetectable data-driven techniques. Despite the collective scientific efforts to identify and mitigate these risks, achieving a thorough comprehension is difficult. These attacks exploit the sensors' vulnerability by utilizing the system's Markov parameters when a particular actuator is compromised, and the detection mechanism is entirely disgusted for this specific covert attack. Furthermore, the literature in [36] investigates undetectable finite-time covert attacks (UFTCA), which provide a distinct set of problems for restricted CPSs. A work on UFTCA design and analysis employs a set-theoretic method and strong controllable arguments to characterize the attack's existence and determine the initial states from which the attack is possible.

In this situation, the attacker gains access to communication channels, knowledge of the plant model, and the capacity to conduct deception attacks to manipulate control signals and make measurements vulnerable.

Table 1.1 demonstrates the strengths and weaknesses of various cyber-attacks as documented in the literature, presenting them in a unified diagram. This allows for a comprehensive overview of the attack scenarios discussed in the literature and aids in identifying existing research gaps.

Related Papers	Attack Type	Attack Strengths	Defender Weakness
[23], [24], [25]	Denial-of-Service (DoS)	High impact potential in critical sectors	Effective mitigation approaches
[26]	Bias Injection	Precision in altering data to induce errors	Incomplete countermeasures against data manipulating
[27], [28], [29], [30]	Replay	Capability to regenerate accurate record for malicious purposes	Limited scope of current detection and prevention methods
[33], [34], [35], [36]	Covert	Expertise in concealing malicious activities within normal operations	Difficulty in detecting and addressing covert

Table 1.1: Summary of cyber-attack types, strengths, and weaknesses.

The investigation into cyber-attack classification has uncovered threats to CPSs, from simple DoS to sophisticated covert attacks. Recognizing and categorizing these threats is effective in finding defence and detection strategies. The next section of the literature review will explore various proposed attack detection mechanisms.

1.2.4 Attack Detection Techniques

Regarding complex and dynamic CPS frameworks, safeguarding against attacks necessitates a deep understanding and innovative detection methodologies. The literature underscores this criticality, focusing significantly on model-based and data-driven detection approaches.

Model-based attack detection involves using pre-defined algorithms or mathematical models to identify anomalies or deviations from expected system behaviour to detect potential cyber threats. Applying methodologies like Chi Square(χ^2), Sliding Mode Observer (SMO), and Unknown Input Observer (UIO) offers robust frameworks for monitoring system status and enhancing attack detection. Traditional or passive model-based detection methods are widely adopted due to their reliance on system models rather than historical data. These methods include χ^2 tests, SMOs, and UIOs, which work by identifying statistical anomalies or model-based deviations without the need to actively interfere with the system's operation. The χ^2 test, for instance, monitors the system for statistically significant deviations from expected behaviour, making it well-suited for detectable

attack scenarios [37–39]. SMO and UIO frameworks provide robust means to handle model uncertainties and disturbances, enhancing detection capabilities. The Adaptive Sliding Mode Observer (ASMO) and the Distributed Sliding Mode Observer (DSMO) are specific examples that have been applied to secure CPS, especially in critical applications such as power systems and DC microgrids [40–43].To address the limitations of passive detection methods, active detection strategies have been developed. These strategies introduce controlled perturbations or external signals to the system, which makes it easier to detect and differentiate attacks, especially those that might otherwise go undetected by passive methods. Examples of active strategies include moving target coding schemes and watermarking techniques, which are added to the system to improve detection in a practical setting.

Moving target coding schemes, as explored in [44–46], involve dynamically altering system parameters to make it harder for attackers to predict or replicate system behaviour, thus complicating replay and covert attacks. Watermarking, another active approach discussed in [47–49], involves embedding a known signal within system communications. This signal serves as a marker for authentic system behaviour and allows for the identification of anomalies induced by attacks that might be otherwise undetectable in a static system. By integrating these methods with traditional model-based detection, researchers have created hybrid detection frameworks that combine passive monitoring with active defence, improving the CPS's resilience to both detectable and previously undetectable attacks.

Both traditional and active model-based approaches contribute to the layered security architecture in CPS by enhancing the system's capacity to detect, identify, and respond to various attacks. While traditional methods remain valuable for their reliability in known attack scenarios, active strategies introduce a dynamic layer that addresses undetectable attacks, offering a complementary solution for more sophisticated threat landscapes. These techniques provide autonomy from historical data, focusing instead on real-time monitoring of system dynamics to flag potential intrusions. Despite these strengths, model-based approaches face challenges, such as dependency on accurate system parameters, difficulties in scalability, and unavailability of high-fidelity mathematical representation of a system, as documented by [50].

Data-driven attack detection has become a cornerstone for identifying cyber-attacks, particularly through machine learning techniques that offer adaptable frameworks for discovering hidden patterns and facilitating decision-making in complex network environments. While effectively monitoring networks with large amounts of data and identifying cyber-attacks, these data-driven methods face challenges in detecting anomalous signals within CPS frameworks [51, 52]. To address this, advancements in data-driven techniques have leveraged both actuator and sensor data, with methods such as Recurrent Neural Networks (RNNs) and Convolutional Neural Networks (CNNs) [53] modelling system behaviour to detect attacks in single CPS units effectively. However, these approaches often struggle with handling complex data from multi-agent subsystems, where command and control centers monitor multiple distributed plant components.

Researchers have explored sophisticated machine learning techniques that do not rely on predefined models, thus enhancing the adaptability detection in CPS. Techniques like hierarchical clustering, deep neural networks, and random forests [54] have been used to identify malicious data points in dynamic environments. Further studies have applied supervised, unsupervised, and semi-supervised learning to improve detection accuracy, as seen in the [55]. [56]. While these techniques provide valuable tools for attack detection, their effectiveness is tempered by the need for large, labelled datasets and the inherent challenges of capturing nuanced system behaviours across diverse subsystems, where data quality is critical [57, 58].

In parallel, the field of intelligent control systems has made notable strides in addressing real-time optimization and adaptability challenges within nonlinear and complex environments. A breakthrough has been integrating reinforcement learning with adaptive control methods, facilitating real-time systems capable of optimizing performance under uncertainty. [59] investigates reinforcement learning applications for designing optimal adaptive controllers, enabling systems to learn and adapt continuously without relying on static, offline designs. Building upon this, [60] introduced a neural network-based actor-critic framework that learns optimal control policies directly from data, a key advantage when system dynamics are partially unknown. [61] demonstrates further progress in this area, which applied reinforcement learning to Linear Quadratic Tracking (LQT) control, enabling optimal solutions even when complete system knowledge is unavailable.

Beyond adaptive control, deep learning and optimization methods have advanced in CPS environments. [62] highlighted the potential of stacked autoencoders combined with simulation-based optimization methods, utilizing algorithms like artificial bee colonies and genetic algorithms to fine-tune structural parameters, thereby enhancing detection accuracy and robustness, as validated through Leave-One-Out Cross-Validation (LOOCV). Concurrently, [63] developed a real-time attack detection framework that adapts to detection delays and false alarms using CNNs and RNNs, achieving precise anomaly detection in sensor networks. This framework represents a significant advancement in adaptable detection systems amid evolving cyber threats, although it introduces complexities in network architecture and substantial training data requirements [57, 58].

Reinforcement learning has also proven effective for anomaly detection and attack resilience in industrial systems. A [64] utilized a Siamese CNN for few-shot anomaly detection, enabling accurate detection even with limited data. This is a crucial feature in environments where attacks, though infrequent, can have severe impacts. Additionally, reinforcement learning has shown potential in managing covert attacks within CPS. For instance, [65] introduced a compensatory controller to mitigate covert attacks, and a [66] employed transformer networks to detect covert attacks by leveraging physical dynamics and non-forgeable stream data. These methods enhance security and offer robust defence mechanisms against sophisticated cyber-attacks, especially in robotics applications.

The literature also explores sophisticated covert attack techniques that challenge CPS security [67] and [68] analyze the mechanisms of covert attacks, focusing on exploiting communication channels and manipulating system parameters [3, 69]. Further investigations at [70]. Also, [71] address covert attack implementation and countermeasures. The challenges associated with time-invariant and time-varying coding techniques for deception attack detection are further discussed in [72] and [73], with the introduction of a two-way coding technique for single-input, single-output (SISO) linear time-invariant systems.

Collectively, these studies represent significant advancements in designing and protecting intelligent control systems. The integration of reinforcement learning, neural networks, and adaptive control strategies has enabled real-time optimization, even under uncertain conditions. Additionally, focusing on detecting and mitigating covert attacks highlights the importance of security in modern CPS. These developments pave the way for more resilient, adaptive, and secure systems capable of

handling the complex challenges of today's interconnected technological landscape.

The current literature on CPS security highlights the critical importance of detecting and mitigating covert cyber-attacks through model-based and data-driven detection methodologies. While advancing knowledge of CPS security, these studies also mention possible limitations in each approach in the table as follows:

Related Papers	**Category**	**Strengths**	**Weaknesses**
[37], [38], [39], [40], [41], [42], [43], [50]	Model-Based	1)Robust frameworks for monitoring system status. 2) Effectively constraining system behaviour, countering disturbances.	1)Reliance on accurate system parameters. 2) Scalability issues.
[54], [55], [56], [57] , [58]	Data-Driven	1)Identifies hidden patterns, facilitates informed decision-making. 2)Demonstrates computational efficiency and adaptability.	1)Complexity in network architecture. 2)Need for substantial training data.

Table 1.2: Comparison of model-based and data-driven detection approaches in CPS security.

Table 1.2 effectively categorized the papers discussed in the context of cyber-attack detection, distinguishing between model-based and data-driven approaches. This overview aids in forming a solid understanding of each approach, clearly delineating their strengths and weaknesses. Such a structured representation is essential in offering a clear viewpoint on these methodologies' distinct characteristics and implications for attack detection. This clarity is crucial for advancing the discourse in this area and guiding future research efforts.

In the comprehensive literature review presented in this master thesis, the complexity and crucial importance of securing CPS against sophisticated cyber threats have been highlighted. Advanced detection methodologies, mainly using coding schemes for identifying covert attacks, have been underscored as areas of significant potential. However, it is known that these coding schemes, while very good at detecting attacks, pose a big problem for researchers: the need to develop stable coding strategies that can be used with both model-based and data-driven approaches. This insight, derived from a meticulous literature analysis, emphasizes the necessity for ongoing, dedicated research endeavours. Such efforts are pivotal in crafting robust and adaptable security measures, safeguarding

CPS against diverse cyber threats, and enhancing their resilience in digitally interconnected configurations.

1.3 Thesis Motivation and Contributions

This thesis offers various contributions due to security concerns of the systems within the CPS framework, focusing on safeguarding against covert and replay attacks. Such undetectable attacks, executed by opponents well-versed in the system's intricacies, pose a significant threat. They allow for the subtle alteration of sensor and actuator data, effectively hiding malicious activities from the detection mechanisms of Command and Control (C&C) systems.

A key innovation in this thesis is developing a coding scheme method that significantly enhances the capability of the χ^2 to detect covert or replay attacks in the model-based approaches. This method employs periodic coding, which remains unreachable to attackers. The plant receives only a one-digit number from the communication channel, enabling it to select information from the predetermined set. This approach, known as the coding-filtering method, represents a novel strategy for identifying attacks that are typically undetectable.

Another main contribution is going through the CPS framework as well, where not all mathematical models and system representations are often available due to security restrictions. In such cases, the only available data are those generated by the C&C inputs and outputs. Here, the detection accuracy becomes paramount, and the thesis focuses on feature selection by random forest and uses a straightforward NN model to reduce the complexity, providing a superior detection capability compared to other methods. This technique will be able to detect covert and replay attacks and identify each attack in concrete scenarios. This approach can effectively present higher accuracy than the other algorithms in the literature.

Together, these contributions address the urgent need for robust and efficient solutions to stealthy attacks in CPS, ensuring these systems' safe and reliable operation, especially in sensitive applications. By introducing the coding-filtering method and innovating advanced neural network techniques, this research marks a significant step forward in the ongoing effort to secure CPS against sophisticated cyber threats.

1.4 Thesis Layout

This master thesis format consists of five chapters that work to detect one of the most sophisticated cyber-attacks on cyber-physical systems (CPS), particularly UAVs. Chapter 1 introduces this famous framework and investigates the advantages and disadvantages completely. By finding the security aspect a main concern in this framework, it explores various historical cyber-attack literature on interconnected systems and attack detection methodologies. It also clearly expresses the problem statement and specifies the significant contributions of this research. Chapter 2 examines various types of attacks and their corresponding detection methods mathematically, provides detailed background on model-based and data-driven approaches, and introduces the complex mathematical model of a nonlinear quadrotor, including its kinematics, dynamic equations, and controller design. Chapter 3 shifts the focus to a model-based detection approach, where a novel coding scheme is designed to augment the effectiveness of the χ^2 in detecting covert attacks. This chapter also discusses implementing periodic coding strategies and secure communication channels to notify the plant regarding the designed coding. In contrast, Chapter 4 takes a data-driven perspective, essential for scenarios that lack a comprehensive mathematical model of the system and rely heavily on data analysis. This chapter critically analyzes the detection and identification processes of attacks, emphasizing the accuracy and reliability of these methods. The exploration terminates with Chapter 5, pooling the results and findings of the preceding chapters. This final chapter provides a comprehensive conclusion of the research and ideas for future work, highlighting potential areas in the field of CPS security that require further exploration and development.

Chapter 2

Background Information

This chapter provides an overview of various types of cyber-attacks mathematically, explicitly focusing on stealthy attacks such as covert and replay attacks within CPSs. Section 2.1 highlights the potential dangers of cyber-attacks and the vulnerabilities in communication networks. The dynamic equations of the nonlinear quadrotor, derived from the Newton-Euler equations, are presented in Section 2.2. The equations presented here include nonlinear control methods for obtaining a stable quadrotor hovering flight and the linearization process around the specific equilibrium point as the basis of this research. This research combines both theoretical and practical aspects of quadrotor flight. Related references regarding model-based attack detections are mentioned in Section 2.3, which investigates different techniques for identifying cyber-attacks using model-based methodologies and evaluates them to determine the most suitable methodology for a certain cyber-attack scenario as the background of this research contributions. Finally, Section 2.5 examines the background of data-driven detection methodologies, emphasizing their significance in addressing cybersecurity issues and the techniques utilized in detecting and identifying cyber-attacks.

2.1 Cyber-attacks Classifications

One of the primary classifications of a cyber-attack is based on the placement of the attack. Cyber-attacks could be accrued through the communication channel and could affect the actuators or/and sensors in the CPS framework, As shown as follows:

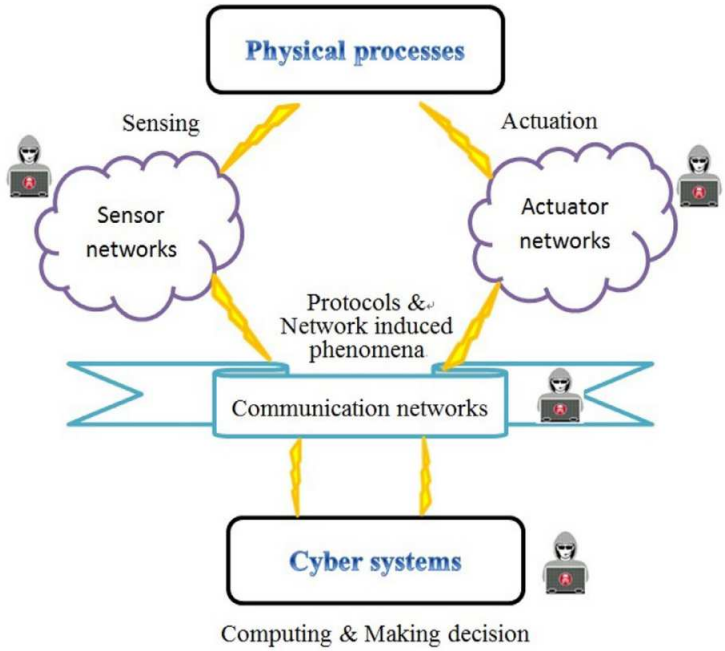

Figure 2.1: Possibility of different types of attacks targeting CPS [4].

As demonstrated in Figure 2.1, attacks pose significant threats to the system's integrity, confidentiality, and availability. These attacks are grouped into principal categories, explored in detail in this thesis section, ensuring an understanding of the systemic vulnerabilities and the corresponding cyber threats. As mentioned earlier, the attack properties can be grouped into system knowledge, disclosure resources, and disruption resources. System knowledge refers to the information about the system that attackers possess, with this knowledge potentially causing the attack. Disclosure resources relate to how confidential information might be exposed and where weaknesses can allow unauthorized access. Disruption resources are associated with the ability to interfere with or halt system operations, with their absence potentially facilitating unauthorized alterations. This framework suggests that cyber-attacks often occur when adversaries leverage deficiencies in these areas to disrupt operations or access data illegally. Figure 1.6 visually represents this classification, detailing the unique aspects of each attack type within this schema and, in some cases, delving into the mechanics of particular attacks. The goal is to thoroughly understand these attacks and evaluate their impact on system security, identifying which ones significantly compromise the system's integrity.

Following subsection goes through the mathematical representation of the system generally and the effect of the attacks on the system particularly.

2.1.1 General representation of the system System

Nonlinear System Model Represents as:

$$\begin{cases} \dot{x}(t) = f(x(t), u(t)), \\ y(t) = Cx(t), \end{cases} \tag{1}$$

where f denoted as a nonlinear function, x, belonging to R^n, represent the state variables, u, part of R^m, signify the system inputs, and y, within R^q, indicate the outputs. The matrix C is a constant matrix used for measurements.

Attack policy [3]: The attack policy, defined as $a_t = g(K, I_t)$, represents a pivotal aspect of the system's vulnerability within the attack space. In this context, K denotes the knowledge of the system available by the attacker, while I_t represents the sensor and actuator data available to the attacker at time t. The variable a_t serves as the attack vector at time t, and it can manipulate system behaviour.

This section examines the system's susceptibility to attack vectors like a_t, offering an analysis of the adversary's system knowledge and the disclosure and disruption resources at their disposal. Specific attacks, such as replay attacks, require minimal system knowledge, underscoring the need to explore disclosure resources. Disclosure resources are crucial for this attacker, enabling them to obtain sequences of data related to control actions and measurements through disclosure resources and disrupt resources by recorded data to pose an undetectable attack.

In addition, the role of disruption resources associated with the specific attack vector is discussed, particularly in their ability to impact system components. This impact may vary based on the attack's nature, whether it involves physical interference, data deception, or a data Denial of Service (DoS). These discussions collectively aim to provide a comprehensive understanding of the adversary model and its components, clarifying the implications and their effects on the security of

control systems.

False Data Injection (FDI) as a Fundamental of the Stealthy Attacks:

False Data Injection (FDI) attacks serve as a foundation for the execution of covert attacks on control systems, where adversaries strategically manipulate system data. These attacks involve the insertion of manipulated signals into the communication channel. In the scenario where additive FDI attacks are present within the input communication channel, aiming to maintain stealth and evade standard detection strategies, a complementary FDI attack within the measurement communication channel becomes a necessary condition. This second FDI attack is required to offset the effects of the initial FDI present in the input channel. The attacks can remain undetectable from conventional detection strategies by employing this dual FDI strategy. This approach not only complicates the detection process but also enhances the stealth aspect of the overall cyber manipulation. [74], [67].

In the additive FDI attack on the input channel, the attack involves adding a manipulated signal $u_a(t)$ to the nominal control input u(t). This attack modifies the control input that the system receives. The Equation (2) representing this scenario in a control system can be expressed as:

$$\dot{x}(t) = f(x(t), u(t) + u_a(t)) \tag{2}$$

The second FDI attack in the output channel involves another additive signal to the actual system output before it reaches the monitoring system. This type of attack is intended to mislead the system about its true state; in this particular case, the signal potentially aims to conceal unauthorized system changes resulting from the first FDI attack. The Equation (3) demonstrate the second FDI attack as follows:

$$\tilde{y}(t) = Cx(t) + y_a(t) \tag{3}$$

The attack signal $y_a(t)$ compromises the immediate reliability and functionality of the targeted systems and challenges the overall trust in system behaviour, underscoring the necessity for robust detection mechanisms that can discern even the most subtle irregularities in system operations. In

Equation (3), $\tilde{y}(t)$ represents the output received by the C&C center.

Figure 2.2 demonstrates the FDI attack on the sensor channel (second phase of FDI) as follows:

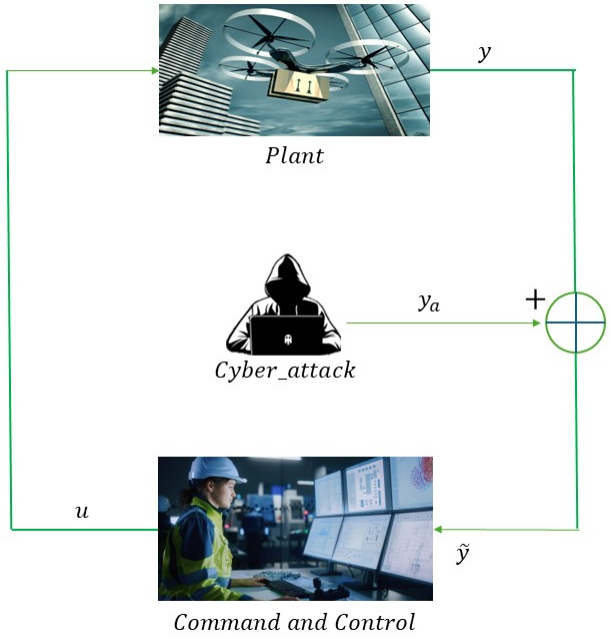

Figure 2.2: FDI attack on actuation channel.

FDI attacks are considered the foundation of stealthy attacks, such as covert and replay attacks. The following sections present the mathematical formulations of these types of stealthy attacks.

Replay Attack

A replay attack requires disclosure and disruptive resources and often exhibits stealth characteristics when targeting systems in steady-state conditions to pose a stealthy attack. It involves recording and replaying a feedback signal with a malicious signal to replace the legitimate one. [75]

The below Figure represents the first phase of the replay attack as follows:

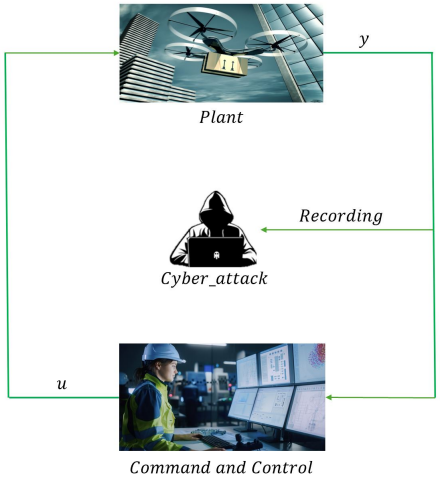

Figure 2.3: First phase of replay attack Configuration

In this step, as shown in 2.3, the attacker only records the data from t_0 until $t_0 + w$. The second phase of the replay attack is shown as follows:

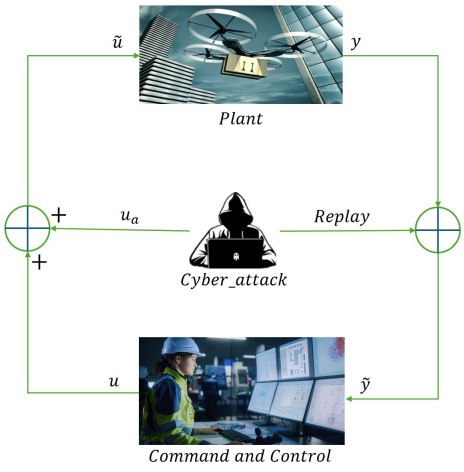

Figure 2.4: Second phase of replay attack Configuration

From t_1, the adversary executes a replay attack by reusing previously recorded data from t_0 until $t_0 + w$ in sensor output data while altering control inputs concurrently. This phase of the replay attack is illustrated in Figure 2.4. In overall, the mathematical representation of the replay attack is shown as follows:

$$\begin{aligned}\dot{x}(t) &= f(x(t), \tilde{u}(t)),\\ \tilde{u}(t) &= u(t) + u_a,\\ \tilde{y}(t) &= y_{(t_0,t_0+w)},\end{aligned} \tag{4}$$

Equation (4) represents the recording and injection phase of the replay attack on nonlinear systems.u_a is the manipulated input regarding the injection phase, and $\tilde{y}(t)$ is the recorded data from the previous interval, by this injection through sensor channel during the injection phase the attack remain stealthy. However, the mathematical representation of the replay attack on the linear systems is shown as follows:

$$\begin{aligned}\dot{x}(t) &= Ax(t) + B(u(t) + u_a(t))\\ \tilde{y}(t) &= y_{(t_0,t_0+w)}(t)\end{aligned} \tag{5}$$

u_a is the manipulated input regarding the injection phase, and $\tilde{y}(t)$ is the recorded data from the previous interval from the linear operation. A, B are considered state space matrices regarding linear systems.

Nonlinear Covert Attack:

A covert attack is a sophisticated maneuver in which the attacker requires the knowledge of the plant model and disruptive and disclosure resources on both actuation and feedback channels. It entails injecting a malevolent vector into the actuation channel to undermine system performance and subsequently introducing another signal into the feedback channel to nullify the initial attack's impact [76]. This type of attack aims to avoid detection by mechanisms deployed on the controller side of the networked control system. Figure 2.5 explains this harmful attack, which continues to be among the most complex attacks discussed in articles, highlighting that there are still unresolved issues in this area requiring further examination.

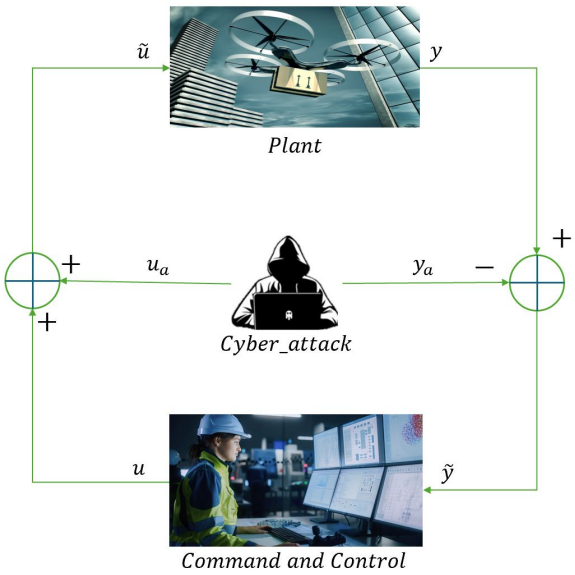

Figure 2.5: Covert attack configuration.

Considering Figure 2.5 and Equation (1), the nonlinear covert attack equations can be described as follows:

$$\begin{aligned} \dot{x}(t) &= f(x(t), \tilde{u}(t)), \\ y(t) &= Cx(t), \\ \tilde{u}(t) &= u(t) + u_a(t), \\ \tilde{y}(t) &= y(t) \pm y_a(t), \end{aligned} \tag{6}$$

The given Equation (6) models the dynamics of a system under the influence of a covert attack. The state of the system, represented by $x(t)$, evolves according to the function f, which is influenced by the control input. In this model, the control input $\tilde{u}(t)$ is expressed as a combination of the standard control input $u(t)$ and adversarial component $u_a(t)$, indicating the effect of the covert attack in the actuation channel. Furthermore, the observed output $\tilde{y}(t)$ from C&C is the output adjusted by an attack-modified output $y_a(t)$, showing how the attack poses stealthy attack by utilizing the system knowledge.

Linear Covert Attack:

In networked control systems, linear covert attacks stand out due to their subtlety and sophistication. These attacks exploit the linear nature of system dynamics, manipulating control signals to degrade performance without triggering detection systems. To pose this attack, the attacker deeply understands the system's vulnerabilities and operational mechanics (system knowledge). The dynamics of a linear system subjected to a basic covert attack can be represented by the following state-space equations [7]:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + B(u(t) + u_a(t)), \\ \tilde{y}(t) &= Cx(t) + D(u(t) - u_a(t)). \end{aligned} \tag{7}$$

In examining a linear system under a basic covert attack, the term $\dot{x}(t)$ is used to represent the time derivative of the system's state vector $x(t)$. The $x(t)$ itself denotes the system's internal states, which collectively determine the system's condition at any moment. The control input $u(t)$, considered the authorized input, is intended to steer the system towards specific set points. In contrast, the attack input $u_a(t)$ is designed by an attacker to degrade the system's performance.

The system is characterized by matrices A and B, which delineate the system's inherent dynamics and the influence of control inputs on the system's state, respectively. The system's output, denoted by $\tilde{y}(t)$, encompasses measurements or observable outputs affected by an attacker. The matrices C and D serve distinct roles in mapping the system's state to its output and representing the direct influence of control inputs on the output, respectively.

Covert attacks constitute a significant and evolving threat, with recent research identifying and analyzing various types. Among these, the Undetectable Data-Driven Covert Attack has attracted considerable attention due to its method of exploiting the system's Markov parameters to reveal vulnerabilities in sensors when specific actuators are compromised, as discussed in [35]. This method highlights the complex interplay between system data and security disruptions, underscoring the challenges of developing comprehensive defence mechanisms.

Further, the concept of Undetectable Finite-Time Covert Attacks (UFTCA) on constrained CPSs, as explored by [36], demonstrates a sophisticated attack strategy that can evade conventional detection methods. However, it is notable that most covert attack strategies, including UFTCA,

have been primarily analyzed within discrete system models. This focus on discrete representations may limit applicability when addressing continuous system dynamics, as is the focus of this research. This discrepancy emphasizes the challenge of translating theoretical attack models and countermeasures into practical defences suitable for real-world systems.

Understanding the impact of stealthy attacks on both linear and nonlinear systems is essential for effectively evaluating defence strategies. To achieve this, one specific system must be examined in detail within the CPS configuration. This approach facilitates a thorough assessment of the proposed strategy's effectiveness. In this research, a quadrotor with six degrees of freedom is selected as the evaluation case study, and the mathematical representation of this system will be presented in detail in the following subsections.

2.2 System Modelling and Quadrotor Specifications

This section highlights the development of a model for the quadrotor. The model includes dynamic components such as hub forces, rolling moments, and changeable aerodynamic coefficients. The realism of quadrotor behaviour is improved, especially during forward flight. Based on Newton-Euler formalism, the model's dynamics offer a thorough foundation for comprehending the behaviour of a rigid body when subjected to external forces. The quadrotor's flight dynamics rely on a combination of momentum and rotor element methods to determine the aerodynamic forces and moments. This method emphasizes the complex and essential equilibrium of forces and moments required for a stable flight.

Figure 2.6 represents the quadrotor's positional and angular orientations, considering its six degrees of freedom.

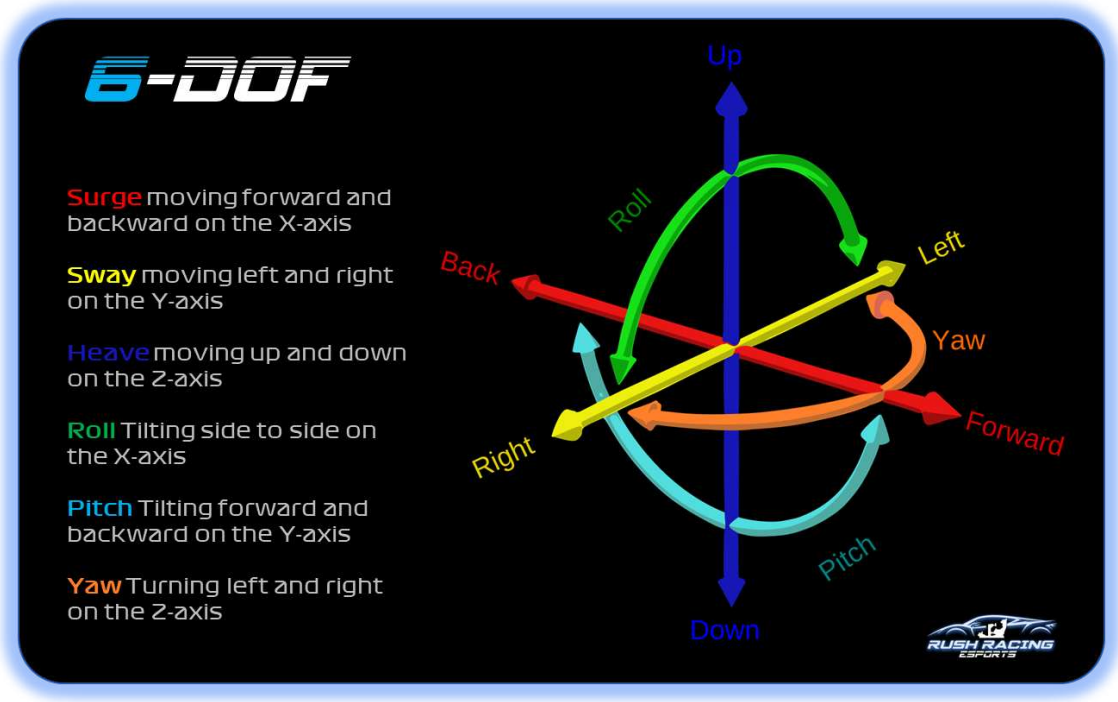

Figure 2.6: Illustration of a quadrotor's motion [5].

Figure 2.6 Demonstrates the quadrotor's six degrees of freedom (DoF) and describes its movement and rotation capabilities in three-dimensional space. Forward and backward movement along the X-axis, termed *surge*, is achieved by tilting the quadrotor forward or backward to direct thrust horizontally. Lateral movement along the Y-axis, known as *sway*, involves tilting the quadrotor to one side. Vertical movement along the Z-axis called *heave*, is controlled by adjusting the rotor's overall thrust; increasing thrust causes the quadrotor to ascend while decreasing it leads to descent. Rotational movements include *roll*, *pitch*, and *yaw*. *Roll* is rotation around the front-to-back X-axis, managed by varying thrust between the left and right pairs of rotors. *Pitch* is the rotation around the side-to-side Y-axis, controlled by differential thrust between the front and back pairs of rotors. Lastly, *yaw*, the rotation around the vertical Z-axis, is achieved by creating torque imbalances through differential rotation speeds of the clockwise and counterclockwise rotors.

The 6-DoF of a quadrotor, which allows it to move and rotate in three-dimensional space freely, is managed by a sophisticated control system. This system adjusts the speed of each rotor independently, enabling the quadrotor to perform complex maneuvers such as hovering in place, vertical takeoff and landing, and even acrobatic flips.

2.2.1 Nonlinear Model with Newton-Euler Formalism

Quadrotors, characterized by their two clockwise (CW) and two counterclockwise (CCW) rotors, are designed with symmetrically positioned rotors for optimal performance and simplified control mechanisms [77].

The following equations briefly describe the dynamics of a quadcopter in flight. The following equations give the quadrotor kinematic model:

$$\begin{aligned}
\dot{x} &= (\sin\phi\sin\psi + \cos\phi\cos\psi\sin\theta)\, v + (\cos\psi\cos\theta)\, u, \\
\dot{y} &= (\cos\phi\cos\psi + \sin\phi\cos\psi\sin\theta)\, v + (\cos\theta\sin\psi)\, u, \\
\dot{z} &= (\cos\phi\cos\theta)\, v + u\sin\theta, \\
\dot{\phi} &= p + r\,(\cos\phi\tan\theta) + q\,(\sin\phi\tan\theta)\,, \\
\dot{\theta} &= q\cos\phi - r\sin\phi, \\
\dot{\psi} &= \frac{r\cos\phi}{\cos\theta} + \frac{q\sin\phi}{\cos\theta}.
\end{aligned} \tag{8}$$

The table 2.1 provides a quick reference for the various parameters involved in the quadrotor's equations:

Symbol	Description
x, y, z	Position coordinates in 3D space
$\dot{x}, \dot{y}, \dot{z}$	Derivatives of position coordinates, representing velocity components
ϕ, θ, ψ	Roll, pitch, and yaw angles
$\dot{\phi}, \dot{\theta}, \dot{\psi}$	Angular velocities in roll, pitch, and yaw
u, v	Control inputs for linear motion
p, q, r	Angular velocities around the x, y, and z axes

Table 2.1: Parameters in the UAV dynamics formulation.

The following assumptions are made for the quadrotor dynamic model [77]:

- The quadrotor's framework remains static.
- Its structure is designed to be symmetrical, with a constant and diagonal inertia matrix.
- The center of mass for the quadrotor is aligned with the body frame's origin.

The control inputs are specifically allocated for the vertical thrust, symbolized by U_1, and for the angular movements, represented by U_2 to U_4. The propellers' cumulative speed is expressed by $\Omega_T(rad/sec)$. This configuration is crucial for precisely regulating the quadrotor's elevation and angular orientation, enhancing its agility and stability during its flight operations [77].

$$U = \begin{bmatrix} U_1 & U_2 & U_3 & U_4 \end{bmatrix}^T \tag{9}$$

The following equations represent the input functions as well:

$$\begin{aligned} U_1 &= b\left(\Omega_1^2 + \Omega_2^2 + \Omega_3^2 + \Omega_4^2\right) \\ U_2 &= b\left(-\Omega_2^2 + \Omega_4^2\right) \\ U_3 &= b\left(\Omega_1^2 - \Omega_3^2\right) \\ U_4 &= d\left(-\Omega_1^2 + \Omega_2^2 - \Omega_3^2 + \Omega_4^2\right) \end{aligned} \tag{10}$$

The dynamic model of the quadrotor is initially described in a body-fixed frame. For control purposes and to generalize the position vector effectively, it is advantageous to express the dynamics of the quadrotor in an earth-fixed frame. Under the assumption that the angular velocity remains consistent across both frames, especially during hovering where the angular transformation matrix T becomes the identity matrix, the dynamics in the earth-fixed frame can be formulated as [78]:

$$\ddot{x} = (\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)\, \frac{U_1}{m}, \tag{11}$$

$$\ddot{y} = (\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)\, \frac{U_1}{m}, \tag{12}$$

$$\ddot{z} = g + (\cos\phi \cos\theta)\, \frac{U_1}{m}, \tag{13}$$

$$\ddot{\phi} = \dot{\mu}\dot{\phi}\frac{I_{yy} - I_{zz}}{I_{xx}} + \frac{J_t p}{I_{xx}}\dot{\mu} - T + \frac{U_2}{I_{xx}}, \tag{14}$$

$$\ddot{\mu} = \dot{\phi}\dot{\psi}\frac{I_{zz} - I_{xx}}{I_{yy}} + \frac{J_t p}{I_{yy}}\dot{\phi} - T + \frac{U_3}{I_{yy}}, \tag{15}$$

$$\ddot{\psi} = \dot{\phi}\dot{\mu}\frac{I_{xx} - I_{yy}}{I_{zz}} + \frac{U_4}{I_{zz}}. \tag{16}$$

Symbol	Description
x, y, z	Position coordinates in the earth-fixed frame
$\ddot{x}, \ddot{y}, \ddot{z}$	Accelerations in the earth-fixed frame
ϕ, μ, ψ	Roll, pitch, and yaw angles
U_1, U_2, U_3, U_4	Control inputs corresponding to thrust and torques
m	Mass of the quadrotor
g	Acceleration due to gravity
I_{xx}, I_{yy}, I_{zz}	Moments of inertia around the x, y, and z axes
J_t	Rotor inertia constant
p	Propeller spin rate
T	Transformation matrix assumed identity during hovering

Table 2.2: Parameters in the quadrotor dynamics.

These equations collectively represent the dynamic behaviour of a quadrotor in flight. Each term corresponds to different physical forces and moments acting on the quadrotor, and their understanding is essential for designing effective control systems. The equations include moving parts for rotation $(\ddot{\phi}, \ddot{\theta}, \ddot{\psi})$, moving parts for translation $(\ddot{x}, \ddot{y}, \ddot{z})$, and the effects of aerodynamic forces and moments, as well as the gyroscopic effects caused by the propellers turning. Combining these

equations forms the basis for simulating and controlling the flight system in various conditions.

This part introduces a model that shows the system's dynamic behaviour in differential equations optimized for control design within the constraints of an embedded control environment. The model emphasizes operational efficiency by optimizing specific dynamic elements, such as hub forces and rolling moments, and standardizing thrust and drag coefficients. This makes it easier to use a state-space representation with a state vector X and an input vector U.

State vector:

$$X = \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} & z & \dot{z} & x & \dot{x} & y & \dot{y} \end{bmatrix}^T \tag{17}$$

$$x_1 = \phi \qquad x_7 = z$$

$$x_2 = \dot{x}_1 = \dot{\phi} \qquad x_8 = \dot{x}_7 = \dot{z}$$

$$x_3 = \theta \qquad x_9 = x$$

$$x_4 = \dot{x}_3 = \dot{\theta} \qquad x_{10} = \dot{x}_9 = \dot{x}$$

$$x_5 = \psi \qquad x_{11} = y$$

$$x_6 = \dot{x}_5 = \dot{\psi} \qquad x_{12} = \dot{x}_{11} = \dot{y}$$

Equation (17) represents the system states which are involved in the mathematical formulation of the nonlinear quadrotor; this equation demonstrates the relationship between the system states as well. However, the nonlinear relation between input and state of the systems is shown as follows:

$$f(X,U) = \begin{pmatrix} \dot{\phi} \\ \dot{\theta}\dot{\psi}a_1 + \dot{\theta}a_2\Omega_r + b_1U_2 \\ \dot{\theta} \\ \dot{\phi}\dot{\psi}a_3 + \dot{\phi}a_4\Omega_r + b_2U_3 \\ \dot{\psi} \\ \dot{\theta}\dot{\phi}a_5 + b_3U_4 \\ \dot{z} \\ g - (\cos\phi\cos\theta)\frac{1}{m}U_1 \\ \dot{x} \\ u_x\frac{1}{m}U_1 \\ \dot{y} \\ u_y\frac{1}{m}U_1 \end{pmatrix} \tag{18}$$

The relationship between input and states considered in Equation (18) delineates essential parameters and control inputs pertinent to the dynamics of a quadrotor system. It outlines both the rotational inertia components, represented by I_{xx}, I_{yy}, and I_{zz}, and the rotor inertia J_r, alongside control strategies that are crucial for the quadrotor's flight dynamics. Parameters a_1 through a_5 are coefficients derived from the moments of inertia, which play a significant role in the quadrotor's rotational dynamics around its principal axes. These coefficients are formulated based on the relationships between the moments of inertia of different axes, ensuring the system's stability and responsiveness.

Furthermore, the control inputs u_x and u_y are articulated to express the quadrotor's motion in the horizontal plane, factoring in the roll (ϕ), pitch (θ), and yaw (ψ) angles, in conjunction with the system's lever arm length (l). These inputs are integral for manoeuvring the quadrotor, facilitating precise positioning and orientation within three-dimensional space. Parameters b_1, b_2, and b_3 relate inversely to the moments of inertia along the respective axes and directly to the lever arm length, translating rotor velocities into linear and angular accelerations. This tabulation is a foundational representation of the quadrotor's physical attributes and control mechanisms, encapsulating the theoretical and practical facets of its flight dynamics.

Table 2.3 demonstrates the Equation (18) parameters and their assigned values:

Parameter	Value	Parameter	Value
a_1	$\frac{I_{yy}-I_{zz}}{I_{xx}}$	b_1	$\frac{l}{I_{xx}}$
a_2	$\frac{J_r}{I_{xx}}$	b_2	$\frac{l}{I_{yy}}$
a_3	$\frac{I_{zz}-I_{xx}}{I_{yy}}$	b_3	$\frac{1}{I_{zz}}$
a_4	$\frac{J_r}{I_{yy}}$	u_x	$(\cos\phi \sin\theta \cos\psi + \sin\phi \sin\psi)$
a_5	$\frac{I_{xx}-I_{yy}}{I_{zz}}$	u_y	$(\cos\phi \sin\theta \sin\psi - \sin\phi \cos\psi)$

Table 2.3: System parameters and control inputs.

2.2.2 Linearization Through Taylor Series Expansion

In pursuing a linearized model, the nonlinear dynamics governing the UAVs are approximated through a Taylor series expansion around an equilibrium point. This expansion, predicated on the assumption of small fluctuations, allows the system to be represented by linear differential equations. The state-space matrices A, B, C, and D are then formulated by evaluating the first derivatives of the system's functions at this equilibrium point, generating a linear model that contains the drone's behaviour close to the steady state. Such linearization is critical for the subsequent application of linear control strategies, which are both robust and computationally efficient for real-time applications. The upcoming steps illustrate the process of linearization using the Taylor series [77].

$$\dot{X} = \left.\frac{\partial f_i}{\partial x_j}\right|_e X + \left.\frac{\partial f_i}{\partial U_j}\right|_e U \tag{19}$$

where $\dot{X}$ is the derivative of the state vector X, $\left.\frac{\partial f_i}{\partial x_j}\right|_e$ is the partial derivative of the function f_i with respect to the state variable x_j evaluated at the equilibrium point e, and $\left.\frac{\partial f_i}{\partial U_j}\right|_e$ is the partial derivative of f_i with respect to the input U_j also evaluated at the equilibrium point e.

- **Assumption 1:** The equilibrium point of the hovering state is considered where the Euler angles are zero.

By utilizing the Taylor Series Expansion concerning the equilibrium point, the following linearized equation around the defined equilibrium point [77]:

$$\dot{x}(t) = Ax(t) + Bu(t)$$
$$y(t) = Cx(t) + Du(t) \tag{20}$$

Consider a system where $\mathbf{x} \in R^n$ is the state vector, $\mathbf{u} \in R^m$ is the control input vector, $\mathbf{y} \in R^p$ is the measured output vector. A, B, C and D are state-space matrices.

2.2.3 UAV's Controller Design

The linearized UAV can be controlled using strategies such as PID Control (Proportional-Integral-Derivative), Linear Quadratic Regulator (LQR), Sliding Mode Control, and fuzzy logic control. Among these, the Linear Quadratic Integral (LQI) method extends the LQR framework by integrating an additional state to compensate for steady-state errors, providing optimal feedback gains for enhanced stability and high-performance system design as shown in [79]. The previous part covered linearizing the mission model and establishing the foundation for defining the controller. As a pivotal element, the controller is designed to play a crucial role in overseeing the system's stability, delivering precise control inputs for the targeted trajectory, and ensuring the reliability and stability of the mission's execution. It is fundamentally responsible for synchronizing the UAV's maneuvers with the established mission parameters amidst the dynamic flight conditions. Figure 2.7 demonstrates the LQI control architecture used in linearized systems as follows:

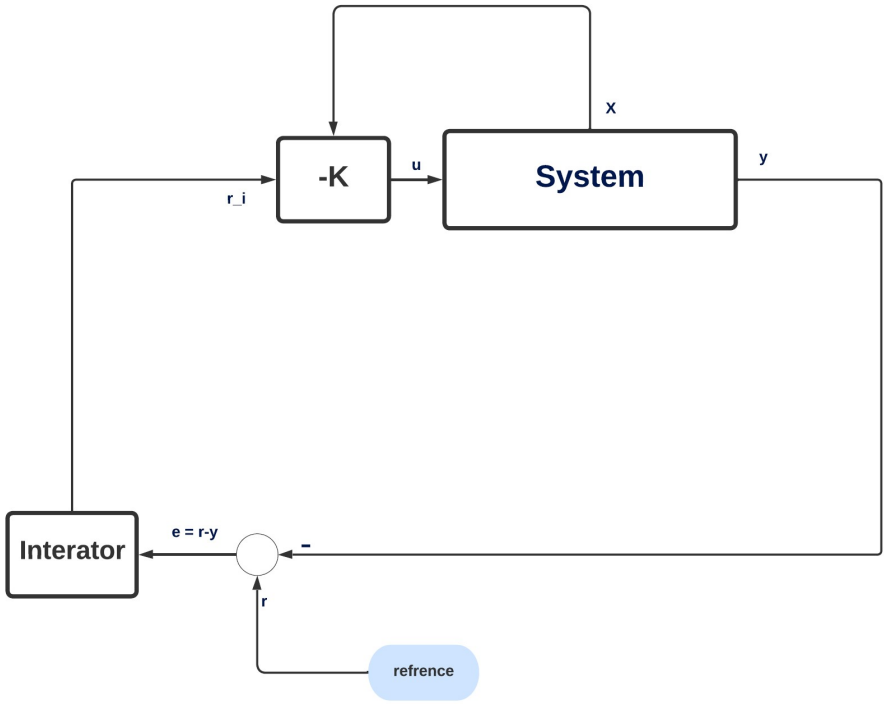

Figure 2.7: Schematic diagram of LQI controller.

The diagram 2.7 presented a Linear Quadratic Integral (LQI) controller configuration schematic. In this configuration, where r_i is the integrator output, system output y and a reference signal r are used to compute the error, which is subsequently integrated to address steady-state errors effectively. The integrated error and the output are processed by a feedback gain matrix $-K$, adjusting the control input u that is fed back into the system. This feedback mechanism ensures the system's output matches the desired reference trajectory.

In the design of the LQI controller, it is first necessary to consider the linearized representation of the system like Equation (20). Typically, the matrix K in the control law $u = -Kx$ governs the relationship between the system states and the control inputs. As an extension of LQR control, LQI control incorporates r_i, which denotes the output of an external integrator and the states, mostly considered as a z that has the effect of both vectors. This formulation ensures that the control law enables the system output y to track the reference signal effectively. In the context of MIMO systems, the number of integrators employed is directly proportional to the dimension of the output y.

This approach guarantees coherent control across multiple system outputs and catches the predefined set point.

The cost function formulation that enables the minimization of K is specified as follows:

$$J(u) = \int_0^\infty (z^T Q z + u^T R u + 2 z^T N u) dt, \tag{21}$$

With tuned Q, R, and N being matrices that weigh the state deviations, control effort, and cross-coupling, respectively. Also, as is shown in the diagram 2.7, $z = \begin{bmatrix} x \\ r_i \end{bmatrix}$ combines the original state x, and r_i is the integral of the error.

The optimal feedback gain matrix K is computed by solving the algebraic Riccati equation under specific conditions [80]:

$$A^T P + PA - PBR^{-1}B^T P + Q = 0, \tag{22}$$

For the tracking problem specifically, part of the augmented state space representation influenced by the integrator effect is considered as follows [81]:

$$z = \begin{bmatrix} x \\ r_i \end{bmatrix}, \tag{23}$$

$$\dot{z} = \begin{bmatrix} A & 0 \\ -C & 0 \end{bmatrix} z + \begin{bmatrix} B \\ 0 \end{bmatrix} u + \begin{bmatrix} 0 \\ I \end{bmatrix} r. \tag{24}$$

By breaking down the Equation (24) can reach the equation as follows:

$$\begin{aligned} \dot{x} &= Ax + Bu \\ \dot{r}_i &= -Cx + Ir \end{aligned} \tag{25}$$

In the augmented state-space representation shown in Equation (24), several matrices play pivotal roles in the evolution of the system's states. The system matrix A and the input matrix B are instrumental in influencing the dynamics of the original states x. Specifically, A determines how the states evolve independently of the inputs, while B modulates the impact of the control inputs

u on the states. In contrast, the output matrix C maps the original states to the integrator states, thereby affecting the calculation of the error integrated in r_i. Furthermore, the identity matrix I is employed to directly transfer the reference input r to the change in the integrator state $\dot{r}_i$, ensuring that the system's response adjusts in accordance with the specified reference trajectories.

Feasibility conditions of the LQI [81]: Several critical conditions must be met to ensure the practical implementation and operational efficacy of the LQI controller. These conditions are designed to affirm the control system's stability, responsiveness, and reliability across various operational scenarios. The stability of the LQI controller hinges on the proper configuration of the system matrices and the characteristics of the control law. As such, the stipulated requirements for the matrices involved in the controller design are not arbitrary but are foundational to achieving a robust and effective control system. These prerequisites are detailed as follows to elucidate the essential nature of each condition in supporting the controller's functionality:

- The pair (A, B) must be stabilizable.
- The matrix R must be positive definite.
- The block matrix $\begin{bmatrix} Q & N \\ N^T & R \end{bmatrix}$ should be positive semi-definite.
- The system $(Q - NR^{-1}N^T, A - BR^{-1}N^T)$ must not have any unobservable modes on the imaginary axis.

This formulation and feasibility conditions ensure that the control system stabilizes the plant and optimally reduces the performance index, allowing the system to track the reference input with minimal steady-state error. This approach ensures that the LQI controller optimizes system performance, tracking the desired output while maintaining robustness and reliability. It effectively synchronizes the UAV's maneuvers with established mission parameters under dynamically evolving flight conditions. In specific frameworks like CPS, the LQI control is integrated into the control system's inner and outer (integrator) loops. This integration allows for broader applicability across various domains.

2.3 Model-Based Attack Detection Approach

Attack detection methodologies are divided into model-based and data-driven approaches, each distinguished by their model construction techniques and analytical development. Figure 2.8 demonstrates the C&C responsibilities such as estimation, observation, detection, mitigation and other critical decision-making activities are centralized within this unit; this unit is separate from the physical system, particularly in the CPS configuration.

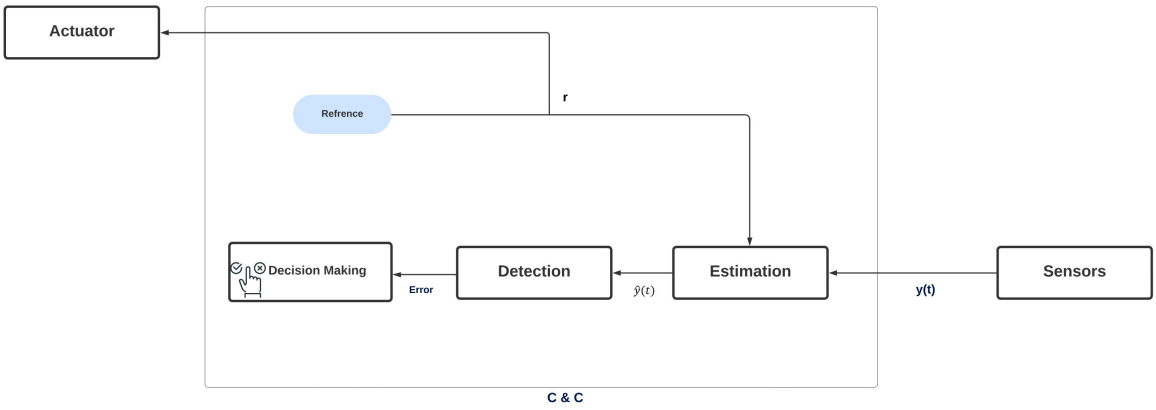

Figure 2.8: Details of the command and control section.

Figure 2.8 shows the C&C structure integral to a system's operational integrity. The diagram highlights the C&C center, which encompasses the continuous estimation of system states from sensor outputs $y(t)$, ensuring real-time accuracy and responsiveness. The reference signal $r(t)$ sets the desired state or outcome. This section investigates the fundamental aspects and distinctions among estimators, observers, and detectors, focusing on the principal methodologies employed by each in the context of attack detection. It will also examine the advantages and disadvantages of these approaches in identifying and responding to cyber-attacks.

As a definition, An estimator refers to any algorithm or method used to infer the values of parameters or states based on observed data. Estimators can be used for a wide range of purposes, not limited to estimating the internal states of a control system. An observer in control systems is specifically designed to estimate the internal states of a system based on its output measurements.

Observers are crucial when not all system states can be measured directly. They use models of the system's dynamics to infer unmeasured states. However, detectors are designed to identify specific events or conditions, such as the presence of a signal noise or an attack or fault. The primary function of a detector is to make a binary or categorical decision based on observed data or signals.

2.3.1 Estimation Methodologies

Estimators and observers share similar mathematical foundations, as noted in their definitions, with their primary distinction lying in the condition of observability that defines an observer. It is emphasized that for a system to be effectively monitored by an observer, its state matrix must satisfy specific observability criteria, ensuring that all system states can be accurately inferred from its outputs. This matrix is constructed from the system's dynamics and output matrices, providing a systematic approach to assessing a system's observability. To understand the concept of observability and its implications in control theory, especially when using estimators and observers, it's essential to consider the mathematical formulations of Equation (20).

Observability Matrix: The observability of a system is determined by the observability matrix $\mathcal{O}$, which is constructed as follows:

$$\mathcal{O} = \begin{bmatrix} C \\ CA \\ CA^2 \\ \vdots \\ CA^{n-1} \end{bmatrix} \tag{26}$$

Here, n is the order of the system (the dimension of the state vector $x(t)$), and A and C are the system matrices from the state-space representation in Equation (20).

Theorem: A system is considered observable if and only if its observability matrix $\mathcal{O}$ has full rank, meaning that the rank of $\mathcal{O}$ is equal to n, the number of states. This condition ensures that all system states can be uniquely determined from the output measurements over time [35].

By ensuring system observability through these mathematical formulations, estimators and observers can effectively perform state estimation, fault detection tasks, and any other threats, enhancing the system's reliability and operational performance amidst complexities and uncertainties.

The following text underscores that the Kalman Filter (KF) stands out among various observation methodologies. It is identified as the most suitable method for the forthcoming study and is presented in the subsequent chapter. The next section will investigate the mathematical underpinnings of the Kalman Filter, shedding light on its efficacy and why it emerges as the preferred choice for the analytical pursuits of this research.

Kalman Filter (KF) Observer

In the domain of control theory, the Kalman filter stands out as a pivotal tool for sequentially estimating the states of linear dynamic systems in the presence of measurement noise. Its derivatives, the Extended Kalman Filter (EKF) and the Unscented Kalman Filter (UKF) extend their application to nonlinear systems, with the specifically designed to avoid the need for linearization. The Particle Filter, renowned for its use of a multitude of particles to depict the state distribution of a system, is particularly suited for tackling complex systems characterized by non-Gaussian noises behaviours. KF is widely recognized for its efficacy in state estimation and is extensively used in attack detection aspects in linear systems [37], [82]. This filter operates through a series of recursive equations aimed at minimizing the mean squared error in the estimation process and is comprised of two primary steps: prediction and update. KF operates through a set of recursive equations.

The prediction step for the system model in Equation (20) is expressed as [83]:

$$\dot{\hat{x}}(t|t-1) = A\hat{x}(t|t-1) + Bu(t) \tag{27}$$

The predicted error covariance is also updated as:

$$\dot{P}(t|t-1) = AP(t|t-1) + P(t|t-1)A^T + Q \tag{28}$$

The update phase incorporates the new measurement $y(t)$ to correct the predicted state estimate:

$$\begin{aligned} K(t) &= P(t|t-1)C^T[CP(t|t-1)C^T + R]^{-1} \\ \hat{x}(t|t) &= \hat{x}(t|t-1) + K(t)[y(t) - C\hat{x}(t|t-1)] \end{aligned} \tag{29}$$

The error covariance is also updated:

$$P(t|t) = [I - K(t)C]P(t|t-1) \tag{30}$$

Here, $\hat{x}(t|t-1)$ is the state estimate at time t given information up to time $t-1$, $P(t|t-1)$ is the predicted state error covariance, $K(t)$ is the Kalman gain, and $P(t|t)$ is the updated estimate of the error covariance.

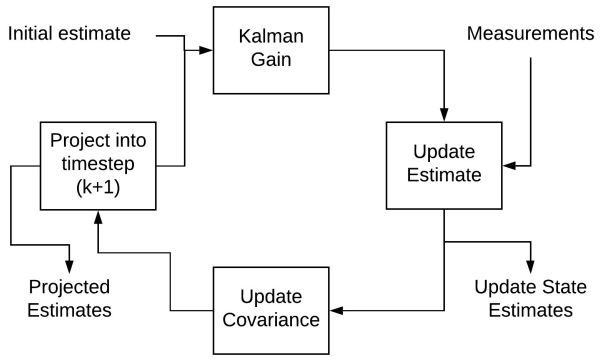

Figure 2.9: Kalman filter diagram [6].

Figure 2.9 illustrates the operational flow of a KF, a recursive algorithm used for estimating the state of a dynamic system from a series of incomplete and noisy measurements. The process begins with an initial estimate of the system's state. This estimate is then projected forward to the next timestep $(k+1)$, resulting in projected estimates. Simultaneously, the kalman gain is computed, optimizing the estimates' weighting based on the system's current uncertainty. The system then incorporates new measurement data to update these estimates, refining them to reflect the system's true state accurately. This update step adjusts both the state estimates and the covariance, representing the estimated accuracy of the state estimates. The updated covariance feeds into the next process cycle, ensuring the system's understanding of uncertainty is continually refined.

2.3.2 Detection Methodologies

The Chi-Square χ^2 detector is a statistical tool used within CPS to detect potential attacks by analyzing the discrepancies between the system's observed outputs and those predicted by a model.

This method is grounded in the principle that, under standard operating conditions, the residuals (differences between observed and predicted values) should conform to a Gaussian (normal) distribution. Deviations from this expected distribution pattern may indicate the presence of faults or cyber-attacks.

The χ^2 test consists of various steps, outlined as follows [84]:

1) **Residuals Calculation:** Residuals are the core component in the Chi-Square detector's operation. For each observation time t, the residual r is calculated as the difference between the actual system output y and the predicted output $\hat{y}$.

$$r(t) = y(t) - \hat{y}(t) \tag{31}$$

2) **Residuals Distribution:** Under normal conditions, these residuals are expected to be small and follow statistical noise with a probability density function equal to that of the normal distribution, commonly known as the bell curve. This is because any well-tuned model of a system should be able to predict the system's behaviour accurately, with deviations primarily due to measurement noise or minor fluctuations in system parameters.

3) **Chi-Square Statistic Computation:** The Chi-Square statistic χ^2 aggregates the squared residuals over a window of N samples, normalized by the residuals' covariance matrix S. This statistical measure captures the magnitude of deviation of the residuals from their expected value (zero for a perfect model under normal conditions).

$$\chi^2(t) = r(t)^T S^{-1} r(t) \tag{32}$$

Equation (32) effectively combines individual residuals, taking into account the expected variance of these residuals. The covariance matrix S adjusts for the fact that not all residuals are equally significant; some may naturally vary more than others due to the system's dynamics.

The χ^2 enables system operators to identify and address abnormalities in real time, preventing potential system failures from the impact of cyber threats. The strength of the Chi-Square detector lies in its statistical specificity and straightforward implementation. However, its effectiveness is

contingent upon the system model's accurate reflection of normal operations and the assumption of Gaussian-distributed residuals. If these requirements are not satisfied, the detector's performance may be affected, requiring the consideration of additional or different detection methodologies. Overall, the Chi-Square detector stands out as a critical element within CPS security and monitoring frameworks, facilitating the proactive identification of attacks through meticulous statistical examination of system outputs. Although these kinds of detection, typically named passive detection methods, are powerful, they cannot identify some sophisticated attacks and need further background knowledge to secure the CPS against these attacks.

2.4 Active Detection Strategy against Covert Attacks

In the evolution of CPS security, integrating some active techniques like watermarking, moving target defence (MTD), and advanced coding designs with traditional detectors makes a robust framework against sophisticated cyber threats like covert and replay attacks. Implementing active strategies within the CPS framework depends on the existing cyber-attack targeted for detection. These critical parameters require careful consideration, particularly when configuring CPS to deploy such approaches effectively. The following section will investigate the background of the modulation matrix design and the advantages and disadvantages of this technique against covert attacks, the most sophisticated attack in the literature.

An active detection strategy using modulation matrices is proposed to counter undetectable attacks, such as covert attacks in [7]. This technique strategically inserts modulation matrices into the system's transfer functions to enhance detection capabilities. However, it is essential to note that while this method provides significant advantages in detecting stealthy attacks, it also introduces potential drawbacks, particularly affecting the system's performance. This section will outline the derivation and implementation of the modulation matrices, followed by a discussion of the advantages and disadvantages of this approach.

2.4.1 Design Strategies

The design of the modulation matrix $S(k)$ enhances the control system's ability to detect and mitigate covert attacks. The strategically designed $S(k)$ matrix alters the system's transfer function, thereby improving the detection of anomalies introduced by the attack signal u_a. This modulation ensures that any deviations caused by malicious activities are more easily identifiable. Furthermore, $S(k)$ aids in maintaining the stability and performance of the system by decoupling the effects of the attack from the desired system behaviour. Integrating $S(k)$ into the control loop demonstrates a proactive approach to safeguarding the system, highlighting its design to enhance the robustness and resilience of the overall control strategy.

The following figure, which is used in [7], illustrates the proposed control system framework designed to detect the effect of covert attacks. This system incorporates various elements, including state variables, modulation matrices, and observer estimates, to enhance the robustness of the control process. The control system dynamically adjusts the input signals and monitors the outputs to identify any deviations caused by potential attacks. By incorporating a modulation matrix and an observer-based control law, the system aims to maintain stability and performance even in the presence of malicious activities. The detailed interaction between the components is depicted in the figure as follows:

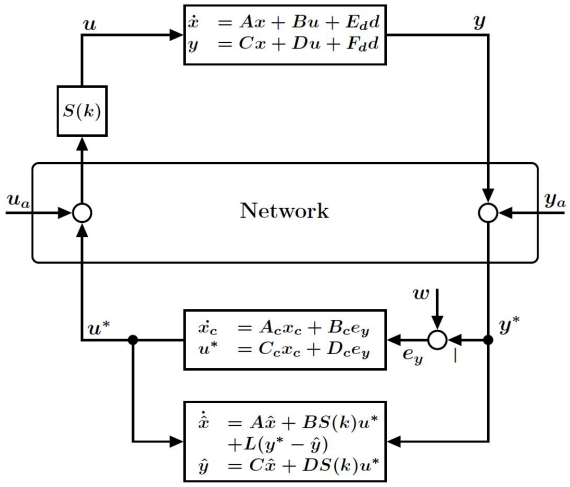

Figure 2.10: Control loop with modulation matrix S(k) [7].

Figure 2.10 illustrates a CPS framework with various parameters and elements. Considering the linear system representation with the unknown disturbance vector and the controller $\mathcal{C}$ on the plant side. The modulation matrix $S(k)$ adjusts the input signal u before it enters the plant, where it combines with the attack signal u_a. The attack affects the output y, resulting in y_a. The observer estimates the states are $\dot{\hat{x}} = A\hat{x} + BS(k)u^* + L(y^* - \hat{y})$ and outputs $\hat{y} = C\hat{x} + DS(k)u^*$. The control law $\dot{x}_c = A_c x_c + B_c e_y$ and $u^* = C_c x_c + D_c e_y$ are driven by the error $e_y = w - y^*$, where w is the setpoint. This framework aims to detect the effects of covert attacks on the system. Considering this general overview, the following part demonstrates the design of the modulation matrix and the effect of this design on the system, step by step.

Constant Modulation

An optimal choice for $S(k)$ would maximize the transfer function $\|G_{ru_a}(s)\|_-$ while minimizing the transfer function $\|G_{yu_a}(s)\|_\infty$. With further conditions, this optimization problem is feasible as driven in [7] in detail. The relationship holds as follows:

$$\max_{S(k)} J_{S(k)} = \max_{S(k)} \frac{\|G_{ru_a}(s)\|_-}{\|G_{yu_a}(s)\|_\infty} \tag{33}$$

Considering Figure 2.10, the transfer function from u_a to r after inserting the modulation matrix $S(k)$, the simplified driven shown in [7] as follows:

$$G_{ru_a}(s) = W \cdot (D + C(sI - A + LC)^{-1}(B - LD))(S(k) - I) = \hat{G}_{ru_a}(s) \cdot (S(k) - I) \tag{34}$$

The gain of an observer matrix L and a weighting matrix W enables the system to perform estimation of the current state $\hat{x}(t)$, prediction of the output $\hat{y}(t)$, and generation of a residual signal $r(t)$. Additionally, $\hat{G}_{ru_a}$ is defined as a transfer matrix, which is designed by an attacker to execute a covert attack. Considering Figure 2.10, the transfer function from u_a to y after inserting the modulation matrix $S(k)$, can be derived as follows:

$$G_{yu_a}(s) = (D + C(sI - A)^{-1}B)S(k) = G_{yu}(s) \cdot S(k) \tag{35}$$

The revise of (33) with respect to Equations (34) and (35) can be written as follows:

$$\max_{S(k)} J_{S(k)} = \max_{S(k)} \frac{\|G_{ru_a}(s)\|_-}{\|G_{yu_a}(s)\|_\infty} \geq \max_{S(k)} \frac{\underline{\sigma}(\hat{G}_{ru_a}(j\omega)) \cdot \underline{\sigma}(S(k) - I)}{\bar{\sigma}(G_{yu}(j\omega)) \cdot \bar{\sigma}(S(k))} \quad \forall\omega \tag{36}$$

Here, $\bar{\sigma}$ denoted as maximum singular value and $\underline{\sigma}$ denoted as minimum singular value. A modulation matrix is designed that satisfies the following boundary conditions for a certain frequency ω_0:

$$\|G_{ru_a}(j\omega_0)\| \geq \gamma$$

$$\|G_{yu_a}(j\omega_0)\|_\infty \leq \rho$$

Considering Equation (34) and the steady state condition for the LTI systems, the optimal S(k) is written as follows [7]:

$$S(k) = c.\hat{G}^{-1}_{ru_a}(s) + I \tag{37}$$

The inverse of the manipulated transfer function is represented by $\hat{G}^{-1}_{ru_a}(s)$, and a scaling factor is denoted by c. When the matrix isn't actively modulating, the system's original behaviour is maintained by adding I (the identity matrix).

The calculation of $S(k) \cdot (u^* + u_a)$ should be carried out quickly and with limited computational power. To address these challenges, limiting the calculations to the frequency $s_0 = j\omega_0 = 0$ is possible. Here is the design of the c in the specific frequency.

$$\|G_{ru_a}(j0)\|_- = c$$

Due to their theoretical advantages, inverse transfer functions like $\hat{G}^{-1}_{ru_a}(s)$ are recognized, but their practical application is often limited by technical and computational constraints. The computation of $S(k)$ is recommended to be confined to zero frequency ($s_0 = j\omega_0 = 0$), simplifying the process to ordinary matrix multiplication. This approach minimizes $\|G_{ru_a}(j0)\|$ to a value c and maintains

$\|G_{yu_a}(j\omega_0)\|_\infty$ within specified limits, enhancing system stability and predictability. The selection of $\omega_0 = 0$ is determined by the strategic necessity to counter covert attacks, which typically introduce disturbances at a steady state, threatening to destabilize the system subtly.

If a constant modulation pattern is maintained, an attacker could identify the system's strategy, allowing the system's modulation pattern to be learned. To counteract this, periodic modulation is proposed, which involves varying the modulation pattern over time. By varying the modulation pattern, it becomes possible to obscure the system's responses from attackers, making it more challenging to predict or effectively interfere with the system.

Periodic Modulation

In contrast to constant modulation where $S(k) = \text{const}$, periodic modulation is defined as $S(k) = S(k+T)$. In a continuous time system context, $S(k)$ represents periodic modulation intervals. This means that k signifies specific time points or periods (e.g., every 3 seconds) during which the modulation matrix changes its value to $S_1, S_2, \ldots, S_T$ in a periodic manner. The mathematical formula is shown as follows [7]:

$$S(k) = \begin{cases} S_1 & \text{for } k = 1 \text{ and } 0 \leq t < t_1 \\ \vdots & \\ S_T & \text{for } k = T \text{ and } t_{T-1} \leq t < t_T \end{cases} \tag{38}$$

Here, $S_1, ..., S_T \in R^{m \times m}$ are constant matrices. The idea is to use each instance of $S(k)$ to identify one channel by making $G_{ru_a}(j\omega) = \hat{G}_{ru_a}(j\omega) \cdot (S(i) - I)$ sensitive to channel u_{ai} in $u_a = [u_{a1} \cdots u_{am}]^T$.while $T = m$.

The problem is reformulated to find a matrix $S(k)$ that rotates the component u_{ai} in the strongest input direction of the residual generator $\hat{G}_{ru_a}(j\omega)$. From previous considerations on directions in MIMO systems, it is known that the solution can be found by considering the SVD of $\hat{G}_{ru_a}(j\omega)$. Since the matrix depends on frequency, the formulation is shown as follows:

$$\hat{G}_{ru_a}(s) = W \begin{bmatrix} I & C(sI - A + LC)^{-1} \end{bmatrix} \begin{bmatrix} D \\ B - LD \end{bmatrix} = W \begin{bmatrix} I & C(sI - A + LC)^{-1} \end{bmatrix} \cdot \beta_D \quad (39)$$

The Singular Value Decomposition (SVD) of β_D is applied because the matrix is efficiently decomposed into its constituent singular values and vectors, revealing the most influential directions of input that affect the system. By the rotation of u_{ai} into the strongest input direction, which corresponds to the highest singular value obtained from the SVD, the input u_{ai} is enabled to excite $\hat{G}_{ru_a}(s)$ effectively across any frequency. This method determines the optimal way to stimulate the system, ensuring a maximum response from the system for given inputs, which is crucial for testing and identifying which channel is under attack. Assume the SVD of β_D is as follows:

$$\beta_D = \begin{bmatrix} D \\ B - LD \end{bmatrix} = U \cdot \begin{bmatrix} \sigma_1 & 0 & \dots & 0 \\ 0 & \sigma_2 & \dots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ 0 & 0 & \dots & \sigma_m \end{bmatrix} \cdot V^H \quad (40)$$

Since V is unitary, all column vectors $v_1, v_2, ..., v_m$ are orthogonal to each other. The strongest input direction is v_1. Thus, $S(k)$ that can guarantee the detection and the identification is chosen as follows [7]:

$$S(k) = c \cdot [s_1 \cdots s_m] + I \in R^{m \times m} \quad (41)$$

Here, $c \in R$ and $s_i = v_1$ if $i = k$ and $s_i = 0$ if $i \neq k$.In this scenario, different matrices can be generated that ensure complexity in contrast to the attacker's knowledge, as well as identify which channel is under covert attack. This procedure effectively enhances the detection procedures against the covert attack and identifies the attack successfully, as noted in [7].

2.4.2 Advantages and Disadvantages

Advantages The partial modulation of the system's transfer functions significantly enhances the detection of stealthy attacks, such as covert attacks, providing a proactive approach to identifying and mitigating potential threats in CPSs. Robust system performance is enabled by focusing on the most significant input directions using Singular Value Decomposition (SVD). Additionally, using the SVD matrix in the design procedures not only aids the detection process but also guarantees the complexity of the matrix regarding security concerns, ensuring the identification of which channel is under attack for future processes. This proposed solution emphasizes the system's ability to adapt and respond to emerging challenges effectively.

Disadvantages: Despite the strategic implementation of periodic modulation, facilitated by SVD, which enhances security, it is recognized that it can adversely affect system performance, particularly in sensitive applications such as UAVs. However, the inherent impact on the system's performance is observed in all cases. This hinders the system's ability to continue its mission post-attack identification and complicates further actions like mitigation. Therefore, meticulous attention is required to ensure that, while detection capabilities are optimized, UAV applications' overall functionality and mission continuity are not compromised.

In the concluding remarks of this section, an extensive overview was provided, encompassing the critical elements pivotal for model-based attack detection. This overview extended beyond traditional attacks found in the literature, highlighting the intricate challenges associated with advanced covert attacks. Particular attention was given to the complexities involved in operating quadrotors, underscoring the imperative need for securing their missions and accurately achieving predetermined set points. The narrative on covert attacks delved into their sophisticated nature and the significant challenges they present to CPS, offering a nuanced understanding of these threats. Additionally, a thorough exposition of the foundational principles and pragmatic approaches to model-based attack detection was presented, showcasing a detailed and applicable analysis. This discussion was further enriched by an exhaustive examination of diverse encryption methodologies specifically tailored to suit the distinct configurations of CPS, thereby providing a solid foundation for defending against the ingenuity of covert attacks. Through this background subsection, not only

was a deeper understanding of potential threats fostered but the system was also equipped with essential tools and strategies to uphold mission security and operational integrity while evolving cyber-attacks.

2.5 Data-Driven Attack Detection and Identification Techniques :

Data-driven approaches become popular in the real world, especially when examining high-fidelity systems. This is due to having full models and mathematical equations of complex systems, where investigating individual subsystems may be impossible. On the other hand, accessing system outputs and sometimes inputs provides a more accurate way to comprehend system behaviour. Neural Network Observers(NNO) are essential in data-driven detection frameworks, providing capabilities beyond traditional model-based techniques. They can perform multiple functions simultaneously as estimators, observers, and detectors in their complex configurations.

2.5.1 Recurrent Neural Networks (RNN)

In recent years, security challenges within these systems have attracted considerable interest, particularly in adopting Neural Network (NN) algorithms. Among these, neural network architectures with multiple layers have demonstrated capabilities in handling both sequential and non-sequential data. When extended over a time dimension, recurrent neural networks (RNNs) exhibit an architecture that effectively adapts to sequences of inputs through an indefinite series of layers. Unlike feedforward neural networks, RNNs utilize their internal states to capture dependencies within input sequences. However, RNNs face notable limitations in their learning process due to the vanishing gradient problem, but they have various advantages compared to feedforward neural networks and work more accurately. A schematic of the unfolded RNN architecture is shown as follows:

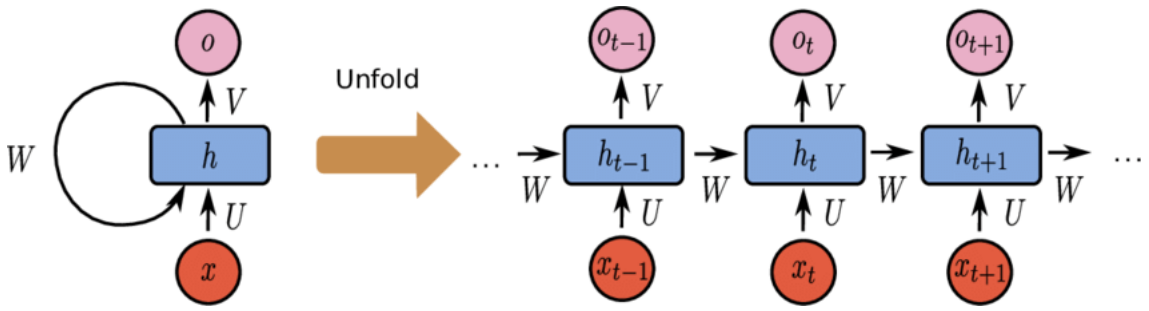

Figure 2.11: Unfolded recurrent neural network. [8].

RNNs can be considered neural networks with memory, retaining information about previously processed inputs. RNNs are highly effective dynamic systems for sequence-based tasks, such as speech recognition or handwriting recognition. Their power lies in their ability to maintain a state vector that implicitly encodes information about the entire history of the sequence. The RNN illustrated in Figure 2.11 makes predictions through a series of matrix multiplications as follows:

$$\begin{aligned} S_t &= f(Ux_t + WS_{t-1}) \\ y &= g(VS_t) \end{aligned} \tag{42}$$

x_t represents the input at time step t. S_t is the hidden state at time step t, functioning as the network's memory. It is computed based on the input at the current step and the previous hidden state. The function f is an activation function that transforms the layer's inputs into its outputs, enabling the fitting of nonlinear hypotheses. Common choices for f include *tanh* and *ReLU*. The initial hidden state, S_{-1}, is typically initialized to zero. The network's output, y, is computed by applying a nonlinear function g, often the *softmax* function, to the matrix multiplication of V and S_t. This function g serves as the activation function for the output layer, converting raw scores into probabilities. Unlike feedforward neural networks, an RNN shares the same parameters across all time steps.

Training Procedures of RNN : There are various methods for training recurrent neural networks (RNNs), including Backpropagation Through Time (BPTT), Real-Time Recurrent Learning (RTRL), and Extended Kalman Filtering (EKF). Feedforward neural networks are typically trained using the backpropagation algorithm. For RNNs, a modified version of this algorithm, BPTT, is employed.

BPTT is implemented by unfolding the RNN over time and stacking identical network copies. Since the parameters to be learned (U, V, and W) are shared across all time steps, the gradient at each output depends not only on the computations of the current time step but also on those of the previous time steps. In RNNs, the cross-entropy loss is a commonly chosen loss function, defined as:

$$L(y_l, y) = -\frac{1}{N} \sum_{n \in N} y_{ln} \log y_n \tag{43}$$

y_l is the number of training examples, y is the prediction of the network, and y_l is the true label. The parameters U, V, and W can be calculated during training by minimizing the total loss on the training data. One popular approach to do this is Stochastic Gradient Descent (SGD). The idea behind SGD is to iterate over all our training examples and, during each iteration, update the parameters in a direction that reduces the error. These directions are calculated by the gradients on the loss function with respect to U, V, and W: $\frac{\partial L}{\partial U}$, $\frac{\partial L}{\partial V}$, and $\frac{\partial L}{\partial W}$. As noted in [85], BPTT can be considered as a black box that gets training data as input and returns these gradients.

One of the powerful RNNs is the Long-Short-Term Memory (LSTM) architecture, in which, despite the RNN's only access to short-term memory, the long-term memory is also available and reachable.

Fundamentals of Long-Short-Term Memory (LSTM) Networks: LSTM networks are a specialized RNN architecture introduced by [85] that excel in learning long-term dependencies. LSTMs can bridge time intervals exceeding 1000 steps, even with noisy, incompressible input sequences, without losing short-term lag capabilities. This is achieved through multiplicative gate units that learn to regulate access to the constant error flow. LSTM networks outperform alternative RNNs, Hidden Markov Models (HMMs), and other sequence learning methods in numerous applications, such as speech recognition and anomaly recognition. The structure of the LSTM is briefly shown as follows:

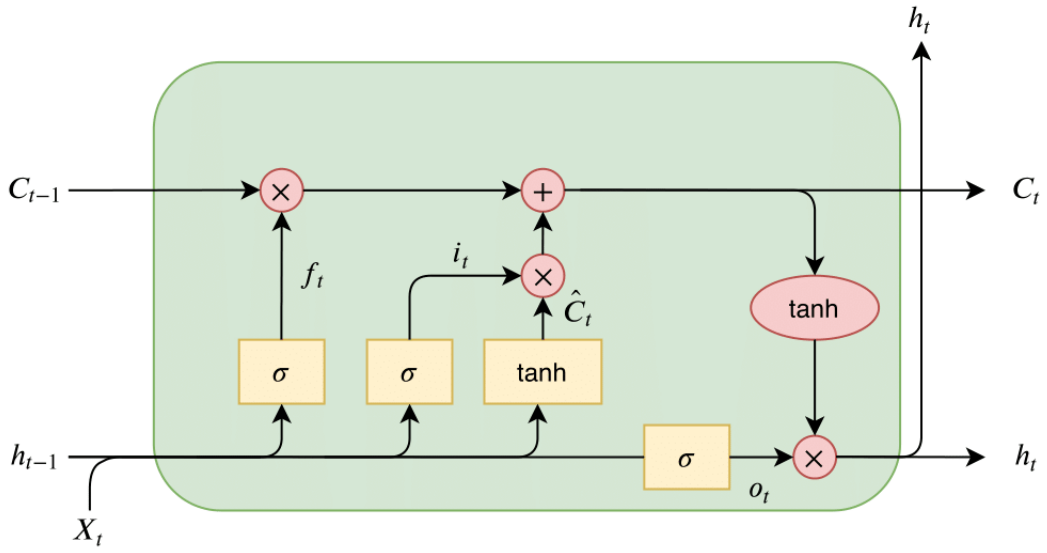

Figure 2.12: Single cell of the LSTM network [9].

Figure 2.5.1 illustrates the structure of an LSTM cell, which is designed to manage information flow over time in recurrent neural networks. The cell processes an input X_t, previous hidden state h_{t-1}, and previous cell state C_{t-1}. It has three main gates: the **forget gate** f_t, which determines what information to discard from the previous cell state; the **input gate** i_t, which controls what new information to add; and the **output gate** o_t, which regulates the information to pass to the next hidden state h_t. The cell combines these gates with the candidate cell state $\hat{C}_t$, computed using a $\tanh$ function, to update the current cell state C_t and produce the new hidden state h_t, enabling the LSTM to retain, update, or forget information as needed over time. The equations concerning each gate of LSTM are shown as follows:

$$\begin{aligned} f_t &= \sigma\left(W_f \cdot [h_{t-1}, x_t] + b_f\right) \\ i_t &= \sigma\left(W_i \cdot [h_{t-1}, x_t] + b_i\right) \\ \hat{C}_t &= \tanh\left(W_C \cdot [h_{t-1}, x_t] + b_C\right) \\ C_t &= f_t * C_{t-1} + i_t * \hat{C}_t \\ o_t &= \sigma\left(W_o \cdot [h_{t-1}, x_t] + b_o\right) \\ h_t &= o_t * \tanh(C_t) \end{aligned} \tag{44}$$

In the above Equations, each gate's function is defined by distinct parameters that shape the flow and storage of information. The σ function represents a sigmoid activation, which outputs values between 0 and 1, effectively controlling the degree of information retention or dismissal at each step. The weights W_f, W_i, W_C, and W_o correspond to matrices that scale inputs x_t and previous hidden states h_{t-1} differently for each gate, while the biases b_f, b_i, b_C, and b_o adjust these outputs further. The forget gate f_t and input gate i_t collaborate to selectively update the cell state C_t by influencing the old cell state C_{t-1} and the newly computed candidate memory $\hat{C}_t$. The candidate memory applies a $\tanh$ activation, yielding values between -1 and 1, thus enabling nuanced adjustments to the cell's stored information. The final output of the LSTM cell, the hidden state h_t, is generated by combining the output gate o_t with the transformed cell state, facilitating the transfer of relevant information to subsequent time steps in the network.

Detection From LSTM Prospective : LSTM networks have found a wide range of applications in cyber-security, where they serve as potent tools for detecting and predicting cyber-attacks in various researches [86], [87], and [88]. These networks are adept at identifying unusual behavioural patterns within systems, interpreting these as indicators of potential attacks. The capability of LSTMs to model temporal information offers a complex understanding of suspicious activities within networks in CPS configuration. This thesis employs LSTM networks to analyze and predict cyber-attacks, utilizing sensors' and actuators' data sources. This method achieves high accuracy in detecting cyber threats by using temporal information within system outputs. Utilizing LSTM networks in cyber-attack detection represents a novel and effective approach. By analyzing temporal patterns in system sensors, these networks can pinpoint cyber-attack patterns, further enhancing the protection of information systems against such complex threats. This integration of LSTM networks into cybersecurity strategies highlights the dynamic intersection of machine learning and information security, especially in the CPS domain, offering a sophisticated framework for safeguarding communication frameworks, as noted in the background.

Disadvantages of LSTM: Although LSTMs offer various advantages, their implementation often requires significant computational resources and incurs substantial costs. Therefore, this research explores alternative techniques that can be applied before deploying the RNN model, aiming to

improve computational efficiency and reduce resource consumption. These techniques include feature selection methods that identify the most relevant features or reveal underlying relationships among them, thus optimizing the input for the main RNN model. By reducing the dimensionality and focusing on key features, these methods enhance the efficiency and performance of the RNN, ultimately making the model more computationally feasible for practical applications.There are different feature extraction techniques as noted [89], [90],and [90] as shown in the figure as follows:

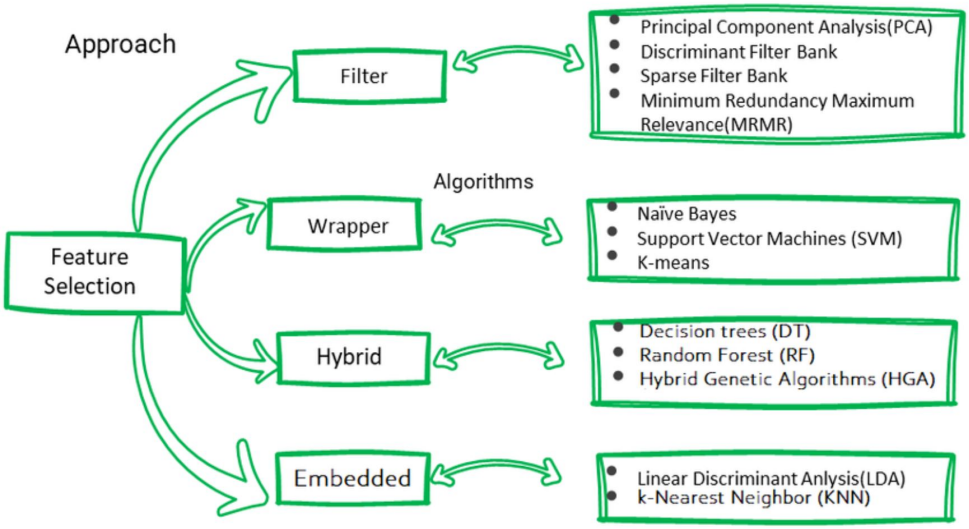

Figure 2.13: Summarize of feature selection techniques [10].

Figure 2.13 categorizes feature selection methods into four main approaches: Filter, Wrapper, Hybrid, and Embedded. The Filter approach evaluates features based on their statistical properties using techniques like PCA and MRMR, independent of any model. Wrapper methods, such as Naïve Bayes and SVM, assess subsets of features by directly training a model, optimizing for performance. Hybrid methods combine elements of both filter and wrapper techniques, utilizing algorithms like Decision Trees and Random Forests to enhance selection efficiency. Lastly, Embedded methods integrate feature selection within the model training process, as seen in methods like LDA and KNN, allowing the model to prioritize features as part of learning.

Feature Importance and Relevance: In feature selection, a common goal is to assess the importance or relevance of each feature with respect to the target variable. This can be quantified in several ways, like measuring the amount of information each feature provides about the target variable. For

a feature x_i and target y, mutual information is defined as:

$$I(x_i; y) = \sum_{x_i \in X, y \in Y} p(x_i, y) \log \frac{p(x_i, y)}{p(x_i)p(y)} \tag{45}$$

where $p(x_i, y)$ is the joint probability distribution of x_i and y, and $p(x_i)$ and $p(y)$ are the marginal distributions. Higher mutual information indicates a stronger dependency between the feature and the target variable. For continuous features, Pearson's correlation coefficient is commonly used to measure the linear relationship between a feature x_i and the target y:

$$\rho_{x_i,y} = \frac{\text{Cov}(x_i, y)}{\sigma_{x_i}\sigma_y} \tag{46}$$

where $\text{Cov}(x_i, y)$ is the covariance of x_i and y, and σ_{x_i} and σ_y are their standard deviations. High correlation values suggest a strong relationship with the target variable.

Generally, there are three approaches for feature selection, each with its mathematical foundation as shown in Figure 2.13. First, filter methods rank features based on statistical scores independent of the model. For instance, Chi-square tests evaluate whether the feature values vary significantly across the categories of the target variable. In the Chi-square test, for example, the test statistic for each feature x_i with target classes C is given by:

$$\chi^2 = \sum_{c \in C} \frac{(O_c - E_c)^2}{E_c} \tag{47}$$

where O_c and E_c are the observed and expected frequencies for class c. Features with high Chi-square values are more likely to be relevant.

Wrapper methods assess subsets of features based on model performance metrics. These methods use sequential forward selection or backward elimination techniques to add or remove features iteratively. The model is trained on each subset, and the subset that maximizes a performance criterion (like accuracy) is selected. As mathematically shown as follows:

$$S^* = \underset{S}{\text{argmax}}\ J(S) \tag{48}$$

Here, $S \subseteq X$ is a subset of features, and $J(S)$ denotes the performance metric. As part of embedded feature selection, the Lasso Regression model incorporates a penalty term into the loss function, which combines the Mean Squared Error (MSE) and a regularization term. The objective function to be minimized is given as follows:

$$\min_{w} \left(\frac{1}{N} \sum_{i=1}^{N} (y_i - \hat{y}_i)^2 + \lambda \sum_{j=1}^{d} |w_j| \right) \tag{49}$$

Here, w represents the feature weights, d is the total number of features, and λ is the regularization parameter. The first term ensures accurate predictions by minimizing the MSE, while the second term penalizes the absolute values of the weights, shrinking irrelevant feature coefficients to zero. As λ increases, the model selects fewer features, balancing predictive performance and model simplicity.

Combining these aspects can be seen as a hybrid technique in feature selection. It evaluates each feature (filter), trains on subsets of features (wrapper), and inherently selects important features as part of model training (embedded). This makes hybrid techniques more powerful and efficient approach for feature selection, as it leverages the advantages of multiple feature selection methodologies, providing both an initial assessment and an iterative refinement of feature importance through its ensemble structure. Decision Trees and Random Forests inherently perform feature selection during training by calculating the importance of each feature in splitting the data. At each split, the algorithm selects the feature that provides the highest information gain or reduction in impurity. In Random Forests, feature importance scores are calculated based on the decrease in impurity across all trees, where a feature is used to split the data. If a feature consistently reduces impurity across different trees and splits, it is assigned a higher importance score.

Given this background, applying these feature selection methods within a CPS framework under cyber-attack conditions presents a promising direction for this thesis. This approach offers a potentially transformative solution, enhancing CPS's security and efficiency. Using feature selection, it is possible to identify the most critical features related to system vulnerabilities, optimize resource allocation and reduce the computational costs of neural network models. This strategy not only strengthens the resilience of CPS against cyber-attacks but also enables more efficient neural

network performance by focusing on relevant data, ultimately minimizing unnecessary calculations and resource consumption.

2.6 Conclusion

As a summary of the background chapter, this foundational section of the thesis provides essential context and understanding necessary for the research milestones ahead. Initially, the chapter covers the chronological development of Cyber-Physical Systems (CPS) and examines the vulnerabilities inherent to this pioneering structure. The focus then shifts to a mathematical representation of specific stealthy attacks that are directly relevant to this research, detailing their characteristics, features, and potential points of exploitation.

Following this, the six degrees of freedom (6DoF) equations of unmanned aerial vehicles (UAVs) are presented, along with the steps for linearization. Given its susceptibility to the types of stealthy attacks under investigation, this UAV model serves as a numerical example throughout the thesis. Subsequently, the structure of CPS and potential threats are discussed, with an in-depth exploration of protection procedures from model-based and data-driven approaches commonly employed in recent literature to enhance CPS security.

This background chapter concludes with an examination of the theoretical and practical foundations needed to address the problem statement outlined in this thesis, with relevant details provided for each main component. With this provided overview, the subsequent chapters will build upon these insights, presenting the novel contributions of this thesis informed by extensive prior research and the groundwork established here.

Chapter 3

Model-Based Detection Method against Covert and Replay Attacks

This chapter is centred on the main contributions to cyber-physical systems, focusing on the control and security of CPSs focusing on UAVs for designated missions. The clarification begins by detailing the process involved in designing a robust control system for a linearized quadrotor model chosen as the primary subject for the study. The details of how the Kalman filter and Linear Quadratic Integrator (LQI) controller work together to ensure system stability while the control centre sends reference signals to the UAV are provided. The interaction of the Kalman filter and LQR controller can effectively regulate the UAV to the desired set points while considering the control effort and system dynamics.

The narrative then transitions to cybersecurity, where the system is prone to covert attack. This attack specifically affected the resilience of the control system. The selection of a chi-square detector for attack detection is justified, with a discussion on its inherent limitations and the reasoning behind adopting a coding matrix to enhance detection capabilities. It is emphasized that this matrix must exhibit a high degree of robustness to remain concealed from potential attackers. The innovative design and implementation of the coding and decoding matrix are subsequently explored. This section highlights how this approach significantly supports the chi-square detector's efficacy in identifying covert attacks. The practical challenges of implementing such a system are also discussed,

focusing on ensuring the coding matrix's security, unpredictability, and confidentiality. Moreover, this designed coding was expanded and tested to detect replay attacks as well.

As the chapter progresses, it addresses the practical challenges encountered during the system's development. These include communication issues due to the distance between the control center and the plant. The chapter also outlines the assumptions made, such as using a secure channel to transmit the chosen matrix index.

This chapter outlines the stand, as Section 3.1 defines the problem. It also includes a configuration that visually demonstrates the existing issue. Furthermore, this section reviews the assumptions made during the problem statement phase, highlighting key factors that frame the problem. Sub-section 3.2 presents the plant-side configuration step by step while the problem is investigated within the CPS framework. Section 3.3 details the first stage of the controller design, focusing on incorporating a Kalman Filter to ensure robust control and accuracy due to regulation problems. This section focuses on cybersecurity issues and presents a highly advanced covert attack designed explicitly for the plant side. This section explains the design and the reasoning behind its sophistication. It is a foundation for future work on tactics for detecting and mitigating the attack. Section 3.3.3 explores the creation of a detection mechanism to identify undetectable attacks, emphasizing the adoption of a chi-square detector and the creative utilization of a coding matrix to improve detection capabilities. Section 3.4 provides a detailed explanation of the design and implementation of the coding and decoding matrix. It specifically examines how this technology enhances security against covert and replay attacks. This addresses the difficulties in guaranteeing the resilience of the matrix while dealing with intelligent attacks. Section 3.5 explores the practicality of implementing control and security measures in real-world scenarios. It covers practical simulations and the difficulties faced during the implementation process. This part thoroughly examines the UAV's performance in several scenarios, demonstrating the efficacy of the devised solutions to detect these two kinds of cyber-attacks. Section 3.6 provides an overview of the chapter's contributions to cyber-physical system security, specifically concerning UAVs and provides valuable perspectives on the undetectable cyber-attack detection domain. Also, some suggestions for future work are mentioned in this part.

3.1 Problem Statement

Security is a top priority in CPSs, especially for systems prone to sophisticated cyber-attacks, such as covert and replay attacks that make the communication channels vulnerable and characterized by their stealth and multi-functionality. Such attacks can disrupt undetected operations, posing significant risks .

As illustrated in Figure 2.5, a covert attack is challenging because it is designed to blend seamlessly with normal system operations, effectively making any injected malfunctions in the output. This makes traditional security measures less effective, as these attacks can bypass standard diagnostic tools. Addressing these kinds of threats is essential to maintaining system security and integrity.

The complexity of stealthy attacks challenges existing cybersecurity strategies. To counter these hidden manoeuvres effectively, detection and mitigation strategies must be rethought. The discussion in this chapter will explore advanced coding-filtering methods, highlighting the necessity of an integrated approach that merges solid system design with cutting-edge cybersecurity tactics to defend against these intricate threats.

In CPSs, decision-making about detecting abnormalities like faults, disturbances, or attacks mainly occurs in the control centre. Sensors and actuators of the system connect this centre through communication channels, which are highly prone to cyber-attacks.

Onboard controllers are crucial for maintaining stability during missions; however, they are insufficient for detecting uncertainties in communication channels caused by cyber-attacks. Despite its indirect access to operational dynamics, the C&C centre is vital in overseeing and addressing potential security threats, underscoring the demand for effective detection methods within CPS frameworks.

Figure 3.1 provides a general overview of the system, illustrating the placement of the controller, estimation and detection mechanisms in the presence of a covert attack. While this configuration does not detail the coding scheme procedure in the detection process or its effects on the detector sector, Figure 3.1 effectively captures how the problem statement is defined as follows:

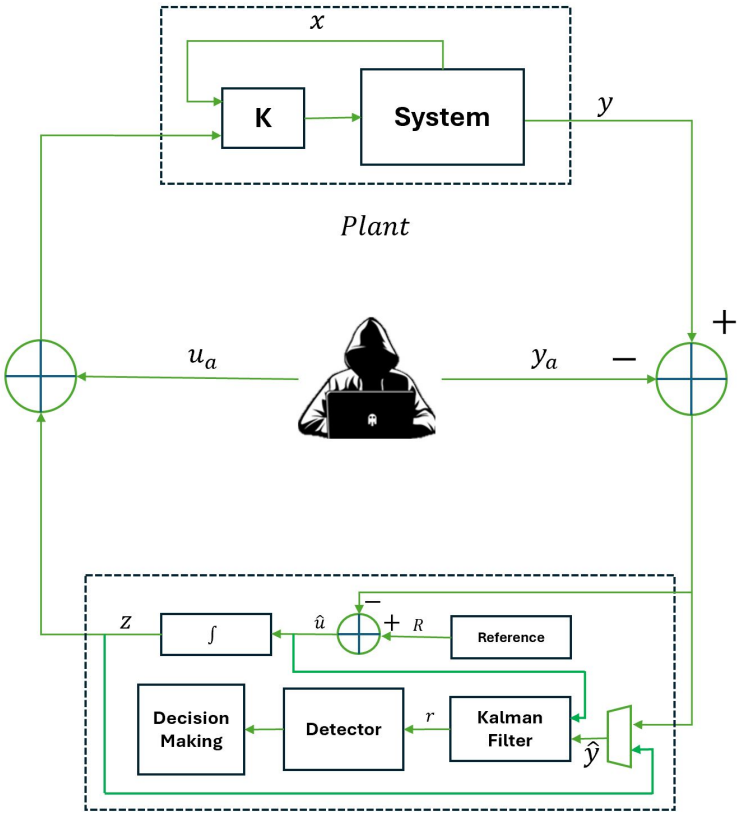

Figure 3.1: Overview of the problem statement.

As depicted in Figure 3.1 for this defined problem, a two-phase controller is presented, which not only ensures system stability but also fulfills set-point regulation objectives, and the different references do not make the system unstable. Additionally, an appropriate estimator is responsible for assessing the system's state, which remains unknown to the C&C center. Moreover, implementing a robust detector is crucial due to the sophisticated attacks on the communication channel. It is important to note that while this configuration does not guarantee detection of such attacks, it is adequate compared to other strategies that can potentially detect them. These procedures will be methodically designed and explained in detail in the subsequent sections.

3.1.1 Problem Definition & Assumptions:

The core issue addressed in this research is the susceptibility of a linear system, set up as a CPS, to covert attacks that traditional security measures detection fail to detect. The sophistication of these

attacks allows them to exploit the system's vulnerabilities while remaining hidden from conventional detection techniques. A refined strategy is proposed to enhance the system's ability to detect these subtle threats. However, the proposed strategy makes certain assumptions and recognizes limitations within the existing problem to make it more manageable and solvable. These assumptions help simplify the complexity while still preserving the overall integrity of the approach.

Assumptions: The first assumption is that no fault is considered during the detection process. Secondly, a secure communication channel from C&C to the plant can send a single encrypted command as outlined by a predefined dictionary, which will be deeply investigated in the coding design. Lastly, the measurements and the process noise are considered through the simulation process, so the detailed mathematical representation did not consider noise with respect to the computation complexity.

Central to the proposed solution is the design of coding matrices, as generally explained in the background chapter. The coding scheme is a potent strategy for enhancing traditional detection mechanisms to identify stealthy attacks like covert ones. Also, the suggested coding will be effective in detecting replay attacks as well. However, a significant challenge lies in mitigating the potential adverse effects these schemes may introduce to the system. The matrix designed for this purpose is pivotal, effectively separating unhealthy data from healthy data in this attack scenario. Ensuring this coding design aligns with the primary detection strategy enhances the system's robustness; also, the confidentiality of this design is provided against these kinds of attacks.

3.2 Step-wise System Representation

As illustrated in Figure 3.1, CPS consists of two primary components: the plant and the C&C, which are interconnected through a communication channel. However, the healthy configuration without an existing attack is shown as follows:

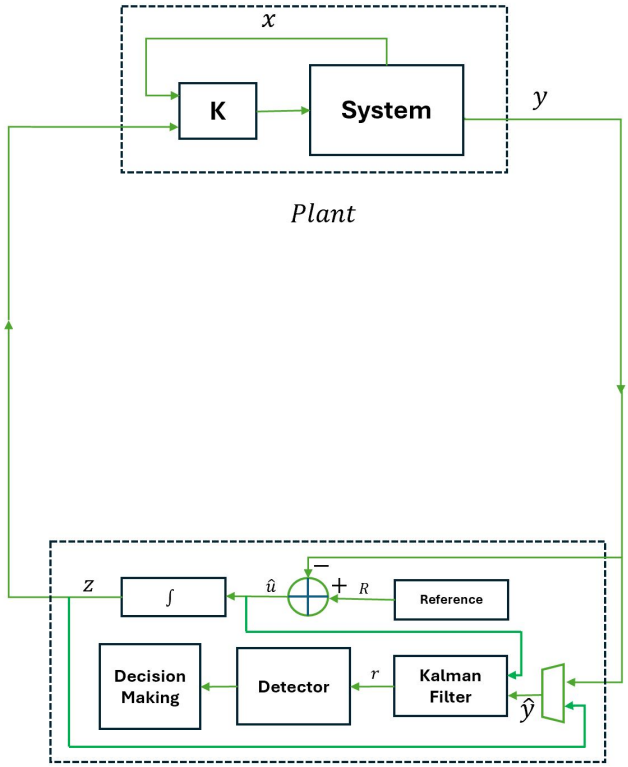

Figure 3.2: Healthy CPS configuration.

A thorough understanding of the mathematical representation of each component is crucial for developing accurate and efficient solutions. The considered plant is modelled as a linear system, which can be mathematically represented as follows:

$$\begin{aligned} \dot{x}(t) &= Ax(t) + Bu(t) + w(t) \\ y(t) &= Cx(t) + v(t) \end{aligned} \tag{50}$$

Consider a system where $x \in R^n$ is the state vector, $u \in R^m$ is the control input vector, $y \in R^p$ is the measured output vector. A, B, and C are state-space matrices. w is the process noise vector. v is the measurement noise vector. w and v are the white noise with the mean of $\mu_w$and μ_v also the variance of $\sigma^2 w$ and $\sigma^2 v$ respectively. As a mentioned assumption, these noises are shown in the simulation result.

As shown in Figure 3.2, a two-phase controller is presented; the considered controller partly takes

place on the plant side, ensuring system stability and the other part is placed in the C&C, fulfilling regulation objective. This consideration reason is respecting the C&C definition as mentioned in the background; these two parts are effectively cooperating together to generate the proper gain. Although the part of the controller exists on the plant side, because of the clarity of the concept, the detail is provided in the C&C design part.

3.3 C&C Design Representation

Regarding CPS configuration, the design of the C&C centre is understood to be crucial. C&C ensures safe guidance and adherence to mission parameters while guarding drones against unauthorized manipulation. As a first step of the C&C, the controller design is discussed in the following subsections, emphasizing its role in addressing stability issues and achieving predefined paths from the C&C center. Additionally, careful attention is required for the design of the estimator, with the Kalman filter being chosen as a robust estimator capable of collaborating with the critical detection component. These elements constitute the fundamental design aligning with CPS.

3.3.1 Dual-Purpose LQI Controller Design:

One of the most powerful controllers suitable for the desired purposes, particularly in the CPSs, is the LQI controller. The required objectives, such as stability and tracking the specific point without causing instability, are addressed by this controller by defining a single control gain, as stated in the background chapter.

Let us revert to Figures 2.7 and 3.2, where the LQI controller is depicted. This particular controller offers various advantages. Placing the gain (K) within the plant side ensures system stability concerning the integral of the error signal. The integrator within the C&C minimizes the error between the desired reference and the actual sensor output. Consequently, through this design, the system can be directed towards specific points. The LQI gain is shown as follows:

$$u(t) = -K \begin{bmatrix} x(t) \\ z(t) \end{bmatrix}, \dot{z}(t) = \hat{u}(t) = R(t) - y(t) \tag{51}$$

The Equation (51) demonstrates the LQI control design concerning the $z(t)$ as integral of the error signal, which is may affected by the attack, and $x(t)$ as the states of the system. By solving the optimized cost function mentioned in Equation (21), the optimal feedback gain that enables the minimization of K can be obtained as follows:

$$K = [K_1^T \;\; K_2^T] \tag{52}$$

Equation (52) represents the nature of the LQI gain, K_1^T is the gain matrix associated with the state vector $x(t)$, which represents the states of the plant. The role of K_1^T is to adjust the control input $u(t)$ based on the system's current state, ensuring desired system behaviour such as stability or performance. K_2^T is part of the gain matrix associated with the state $z(t)$, which typically represents additional dynamic states such as estimation errors. In the provided control law, $z(t)$ is driven by the difference between the reference signal $R(t)$ and the measured output $y(t)$, thus acting as a correction or compensation term in the control input. The LQI controller augments the existing plant equations.

In this framework, the augmented representation of Equation (50) is formulated as follows:

$$\begin{aligned} \dot{\hat{x}}(t) &= \hat{A}\hat{x}(t) + \hat{B}\hat{u}(t) + \hat{w}(t) \\ \hat{y}(t) &= \hat{C}\hat{x}(t) + \hat{v}(t) \end{aligned} \tag{53}$$

Where the augmented state vector, matrices, and disturbance terms are defined as follows:

$$\hat{x} = \begin{bmatrix} x \\ z \end{bmatrix}, \quad \hat{A} = \begin{bmatrix} A & 0 \\ 0 & I_{m\times m} \end{bmatrix} - \begin{bmatrix} B \\ 0 \end{bmatrix} K, \quad \hat{B} = \begin{bmatrix} 0 \\ I_{m\times m} \end{bmatrix},$$

$$\hat{C} = \begin{bmatrix} C & 0_{p\times m} \\ 0_{m\times n} & I_{m\times m} \end{bmatrix}, \quad \hat{w} = \begin{bmatrix} w_{n\times 1} \\ 0_{m\times 1} \end{bmatrix}, \quad \text{and} \quad \hat{v} = \begin{bmatrix} v_{p\times 1} \\ 0_{m\times 1} \end{bmatrix}.$$

Here, the augmented state vector $\hat{x}$ combines the original state x with additional dynamic states z. The augmented system matrix $\hat{A}$ includes the original dynamics matrix A and an effect of the close loop control gain matrix K. The control input's effect is introduced through the matrix $\hat{B}$,

which specifically impacts the integral state z, and the output matrix $\hat{C}$ captures measurements from both the plant outputs and the integral states, allowing the controller to monitor and respond to all relevant system components. Lastly, vectors $\hat{w}$ and $\hat{v}$ are extended to match the augmented system dimensions, where $\hat{w}$ represents augmented process noise and $\hat{v}$ accounts for augmented measurement noise.

While direct access to the system states is unavailable, implementing an appropriate estimator is essential for estimating and evaluating the system states and outputs against the actual output generated from the plant side. This evaluation helps to identify any abnormalities that may occur during data transmission through the communication channel. Such insights are crucial for keeping the system secure and promptly addressing discrepancies or errors.

3.3.2 Kalman filter Design:

The Kalman filter is often chosen as an observer due to its significant advantages, as Section 2.3.1 mentions. It is crucial for maintaining high reliability and accuracy in systems where direct observation of states is unfeasible. Its recursive nature permits the continuous refinement of state estimates with incoming data, ensuring adaptability and efficiency by minimizing computational resources, optimal estimation, and noise handling. This makes the Kalman filter particularly suitable for real-time applications, where it balances performance with minimal computational resource demands. As mentioned in the assumptions, noise was not considered in the covert attack detection procedures. However, incorporating noise is essential for simulation results, mainly as this research is focused on application-based sectors. This consideration underscores the rationale for choosing the Kalman filter.

As detailed in the background of the Kalman filter in Section 2.3.1, let us revisit the conditions that must be met by Equation (53), outlined as follows:

$$\begin{aligned} &E[\hat{w}(t)] = E[\hat{v}(t)] = 0, \quad E[\hat{w}(t)\hat{w}^T(t)] = Q(t), \\ &E[\hat{v}(t)\hat{v}^T(t)] = R(t), \quad E[\hat{w}(t)\hat{v}^T(t)] = N(t) = 0 \end{aligned} \tag{54}$$

Here, E represents the expected value, indicating the mean values of the augmented process noise

$\hat{w}(t)$ and the augmented measurement noise $\hat{v}(t)$, which are assumed to be zero, denoting unbiased noise characteristics. $Q(t)$ is the covariance matrix of the augmented process noise, and $R(t)$ is the covariance matrix of the augmented measurement noise. The term $E[\hat{w}(t)\hat{v}^T(t)] = N(t) = 0$ indicates that the augmented process noise and the augmented measurement noise are uncorrelated, represented by a zero cross-covariance matrix.

Regarding Figure 3.2 and Equation (53), the equation of the Kalman filter is clarified as:

$$\begin{aligned} \dot{\hat{x}}_o(t) &= \hat{A}\hat{x}_o(t) + \hat{B}\hat{u}(t) + L(\hat{y}(t) - \hat{C}\hat{x}_o(t)) \\ \hat{y}_o(t) &= \begin{bmatrix} C & 0 \end{bmatrix} \hat{x}_o(t) \end{aligned} \tag{55}$$

Here, the variable $\dot{\hat{x}}_o(t)$ represents the estimation of the augmented state vector $\hat{x}_o(t)$, an observer gain matrix, L. The observer gain $L = \begin{bmatrix} L_1 & L_2 \end{bmatrix}$ is split into L_1 and L_2 to provide tailored feedback for different parts of the augmented state vector. Precisely, L_1 adjusts the estimation for the primary plant states, while L_2 focuses on the additional integral or error states. Finally, $\hat{y}_o(t)$ represents the estimated system output, as shown in the equation C&C does not have access to the $\hat{C}$. With a detailed explanation provided for the Equation (55), the only parameter requiring a precise definition is the matrix L, which serves as the gain of the KF. The gain is formulated as follows [91]:

$$L = P\hat{C}^T(\hat{C}P\hat{C}^T + R)^{-1} \tag{56}$$

Where $P(t)$ is the error covariance matrix, $\hat{C}^T(t)$ is the transpose of the output matrix $\hat{C}(t)$, and $R(t)$ is the measurement noise covariance matrix. The minimized $P(t)$ can be achieved by solving the Riccati equation as follows:

$$\dot{P} = \hat{A}P + P\hat{A}^T + Q - P\hat{C}^T(\hat{C}P\hat{C}^T + R)^{-1}\hat{C}P \tag{57}$$

In the steady-state situation, $\dot{P} = 0$, leading to the algebraic Riccati equation:

$$0 = \hat{A}P + P\hat{A}^T + Q - P\hat{C}^T(\hat{C}P\hat{C}^T + R)^{-1}\hat{C}P \tag{58}$$

Where Q is the process noise covariance matrix. These equations collectively ensure the Kalman filter dynamically adjusts the state estimates based on the latest measurements while minimizing the estimation error in the steady-state scenario. The choice of the minimal P and the optimal L ensures that the Kalman filter can estimate the states accurately, ultimately achieving steady-state convergence.

A detailed understanding of the existing configuration in a healthy scenario is provided regarding Figure 3.2. In this context, Figure 3.1 is utilized to illustrate the equations and characteristics associated with the existing attack on the augmented system. The equation of the augmented system under a covert attack is represented as follows:

$$\begin{aligned} \hat{\dot{x}}(t) &= \hat{A}\hat{x}(t) + \hat{B}\hat{u}(t) + B_a u_a(t) + \hat{w}(t) \\ \hat{y}(t) &= \hat{C}\hat{x}(t) - Ey_a(t) + \hat{v}(t) \end{aligned} \tag{59}$$

The matrix $B_a = \begin{bmatrix} B \\ 0_{m\times m} \end{bmatrix}$ is specifically structured to direct the attack input $u_a(t)$ to the appropriate state variables, introducing vulnerabilities that impact the system's behaviour. Additionally, the measurement equation incorporates the matrix $E = \begin{bmatrix} I_{p\times p} \\ 0_{m\times p} \end{bmatrix}$, which allows the attack signal $y_a(t)$ to directly affect the measured output $\hat{y}(t)$. This term effectively alters the output measurements by subtracting the attack influence $Ey_a(t)$, making it challenging to detect the covert nature of the attack. However, the attack model for more clarification is shown as follows:

$$\begin{aligned} \dot{x}_a(t) &= (A - BK_1)x_a(t) + BK_2u_a(t) \\ y_a(t) &= Cx_a(t) \end{aligned} \tag{60}$$

Equation (60) represents the attack model, mathematically expressing the extent of system knowledge accessible to the attacker. This knowledge allows the attacker to exploit vulnerabilities within the system while maintaining undetectability from the defender.

3.3.3 Designing Detection Methodologies

The Chi-square detection approach analyzes residuals and deviations between a system's actual outputs and an observer's estimated outputs from the Kalman filter. This method is highly effective because it can systematically evaluate the importance of observed deviations, providing a highly accurate statistical threshold for differentiating between typical variations caused by unpredictability and possible attack signals. The two main steps in this part are designing the χ^2 method for the specifically defined problem and then defining a valid threshold that can effectively detect any abnormal behaviour in the system.

Chi-square Detector Design:

To begin the first step, the definition of the residual is required. Residual $r(t)$ is the difference between the actual system output, which may be under attack, and the estimated output from KF, based on estimated states, which is mathematically explained in Equation (55). The formulation of the residual is mathematically represented as follows:

$$r(t) = \hat{y}_o(t) - y(t) + y_a(t) \tag{61}$$

The term $\hat{y}_o(t)$ represents the estimated output derived from the estimator, while $y(t)$ denotes the actual output of the system under attack. The variable $y_a(t)$ captures the elimination phase of the covert attack, contributing to the attack's undetectability by the C&C center. As shown in Equation (59), the estimated output is formulated using a portion of the matrix $\hat{C}$. However, the attacker utilizes matrix C, which is based on system knowledge without the influence of augmentation.

The core of the Chi-square detection strategy lies in calculating the Chi-square statistic, which

normalizes the magnitude of residuals against their expected variance as given by:

$$\chi^2(t) = r(t)^T S^{-1} r(t) \tag{62}$$

S is the covariance of the residual from Equation (61). This statistic offers a standardized metric for evaluating deviations from the norm and ensures that the residuals primarily reflect inherent system noise, providing a baseline for detecting anomalies, which is here specifically investigating the attack.

Defining the Detection Threshold:

As the second step for the detection procedure, the Monte Carlo method is a powerful statistical technique widely recognized for its effectiveness in complex simulations, particularly in generating critical thresholds for systems like Chi-square detection in attack identification. Randomly simulating a vast array of scenarios that mimic the system's normal operational conditions enables the precise calibration of detection thresholds, such as γ, ensuring an optimal balance between sensitivity and false alarm rates. Its utility lies in its ability to model the unpredictable nature of real-world data and disturbances, making it an invaluable tool in the rigorous assessment and enhancement of detection strategies within various domains.

Monte Carlo Procedures:

(1) **Simulation of Normal Operation:** Generate numerous residual samples $r(t)$ based on the assumed noise characteristics and system model under normal conditions. Compute $\chi^2(t)$ for each sample using the Equation (62).

(2) **Distribution Analysis:** Analyze the distribution of $\chi^2(t)$ values to establish the behaviour under the null hypothesis.

(3) **Threshold Setting:** Determine γ such that the probability of $\chi^2(t)$ exceeding γ under normal conditions equals the desired false alarm rate (α):

$$P(\chi^2(t) > \gamma | \text{healthy conditions}) = \alpha \tag{63}$$

This probability α is typically set to a small value, such as 0.05 or 0.01, indicating a 5% or 1% false alarm rate, respectively.

Advantages of the Monte Carlo Method

The Monte Carlo method offers flexibility in modelling complex systems. It allows for customization based on system-specific characteristics and provides accurate estimates for γ by simulating a wide range of conditions.

By implementing the Monte Carlo method alongside the Chi-square detection strategy, the designed system can effectively calibrate a detection threshold that optimizes the balance between sensitivity to any systematic abnormalities and minimizes false alarms. This approach significantly enhances the system's robustness, thereby maintaining system integrity and performance.

The carefully designed system control and attack identification procedures are solid and powerful measures. However, the advancement of attackers poses significant challenges to such detectors, necessitating a proactive approach to stay ahead. While these methods hold promise for identifying attacks, their efficacy hinges on anticipating sophisticated threats, such as covert attacks, highlighted in the problem statement. Success in countering these threats requires the integration of specialized parameters that enhance the detector's capability to identify such nuanced attacks, ensuring the defender's triumph in this confrontation. The subsequent section will delve into the coding scheme procedure, outlining the essential conditions and parameters designed to bolster this sophisticated detection strategy, thereby providing a cohesive conclusion to the detection segment and underscoring its criticality in maintaining system security and integrity.

3.4 Design Principles of Coding Matrix

The coding scheme is a strategic approach embedded within the system's operation to enhance stealthy attack detection capabilities. It involves inserting specific signals or codes and analyzing the system's response to these codes against a predicted response based on the system model. This strategy encompasses estimation techniques, predicting the expected system behaviour in response to the coded signal, and active detection techniques, involving introducing the code into the system's

operations. The coding scheme procedure must address several critical factors to ensure effectiveness and stealthiness. Firstly, the optimal placement of the coding scheme within the system is paramount to keep it concealed from potential attackers, ensuring that it does not appear as a recognizable system state and remains unaffected by deliberate manipulations to neutralize its impact on the output. Secondly, the strategy's influence on the system's states should be carefully considered because of an appropriate decoding mechanism before input injection to prevent any adverse effects on the system's functionality. Thirdly, the complexity of the designed matrix is a significant concern; it must possess sufficient sophistication to bypass detection by the attacker while remaining reliable and effective throughout the mission's duration. This section will methodically explore these concerns, providing a step-by-step mathematical clarification of viable solutions that comprehensively address these issues, ensuring the coding scheme's integrity and efficacy.

3.4.1 Design Procedures:

As previously noted, ensuring the optimal placement of the coding scheme within the system is crucial for concealing it from potential attackers. This placement guarantees that the coding scheme does not become a recognizable part of the system state and remains protected from intentional manipulation to neutralize or diminish its effect on the output. The system's security is enhanced by strategically positioning the coding-decoding process. The coding and decoding matrix placement is illustrated as follows:

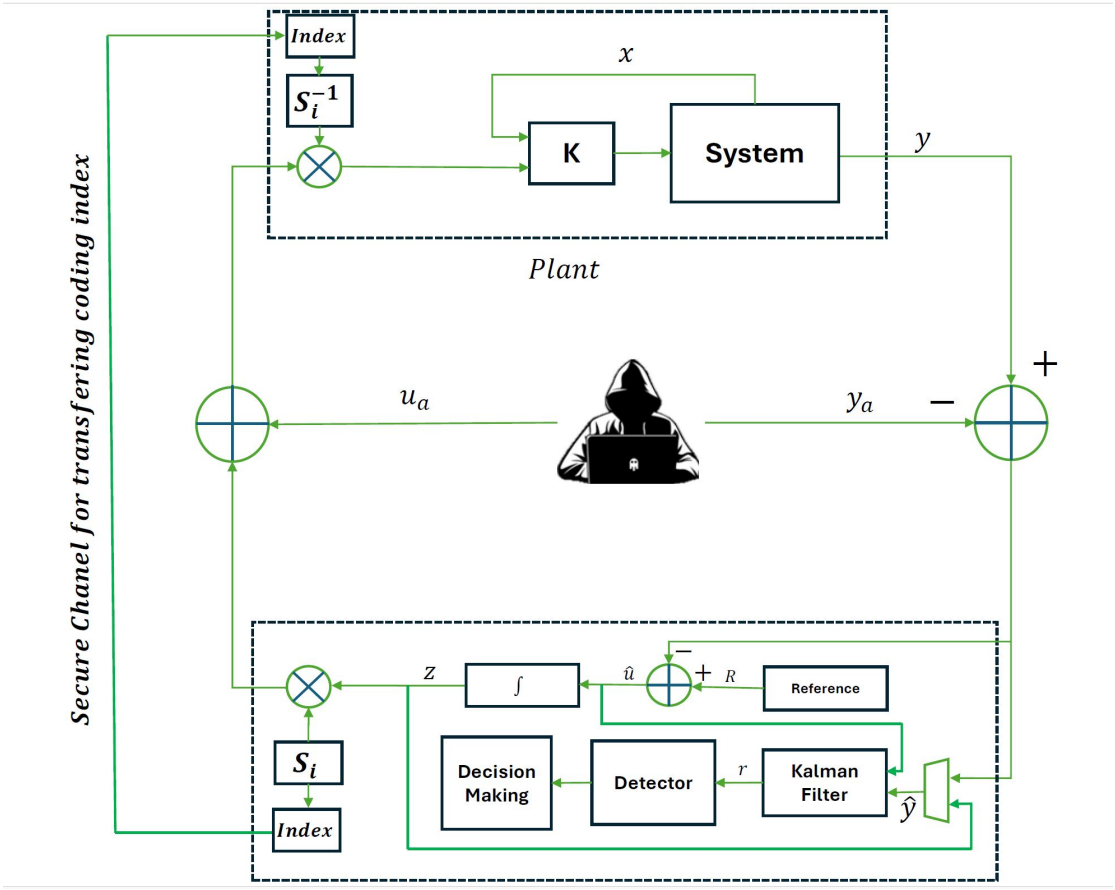

Figure 3.3: Coding & decoding placements in the CPS.

Figure 3.3 visually demonstrates the proposed strategy for the detection problem against covert attack. The figure illustrates the placement of the coding matrix S_i in the C&C side, where it is multiplied with the generated command. The decoding process occurs on the plant side before any action by S_i^{-1}. This placement not only helps counter this specific cyber-attack, even if the attacker attempts to eliminate its effect on the output, but also perfectly guarantees the system's functionality by decoding. Additionally, the presence of a secure channel in the figure is notable. This secure channel is intended to transfer a digit securely, referred to as the decided coding matrix index, which will be investigated following.

The mathematical basis for how this placement aids the detection process and which matrix design strategy offers the most benefits for subsequent steps will be investigated.

According to Theorem 3 from [35] if there exists an invertible matrix T_q such that the rank of

$CA^nBT_qL_q = 1$, where C, A, and B are system matrices, T_q is the coding matrix related to the specific actuator, and L_q is the attack signal on the specific actuator, the detection goal can be accomplished.

While any coding matrix satisfying this condition can be considered valid, notable disadvantages exist. It is mentioned in their research that only one coding matrix T can be utilized for all actuators simultaneously, as using multiple matrices would complicate the decoding process. Implementing different decoding processes for each actuator would significantly increase complexity, making the system more difficult to manage and potentially less reliable. Thus, the necessity of using a single matrix T simplifies implementation but limits the coding strategy's flexibility, potential reliability, and effectiveness.

In the chosen research, the impact of the decoding matrix is more significant. Any invertible matrix could fulfill the role of both coding and decoding. The decoding procedure design will be examined first to gain further design advantages. This perspective allows for a more effective decoding process, directly influencing detection and identification processes.

Design Procedures of Constant Coding Matrix:

Given the above theorem and the discussion in Section 2.4.1 of the background, the objective is redefined to concentrate on designing matrices that ensure the detection mechanism and desired complexity remain unknown to attackers and assist in the identification problem. This involves demonstrating which sensors and actuators are under attack.

Let us initially derive the transfer function with respect to Figure 3.3, while considering S_{i_c} as a constant coding matrix in this phase of representations, which links the attack's input signal u_a to y, as is represented as follows:

$$\begin{aligned} G_{yu_a}(s) &= (I - (C(sI - (A - BK))^{-1}B)M(s))^{-1}(C(sI - (A - BK))^{-1}B)S_{i_c}^{-1} \\ &= G_{\tilde{y}\tilde{u}}(s)S_{i_c}^{-1} = T(s)S_{i_c}^{-1} \end{aligned} \tag{64}$$

Here $\tilde{y}$ is the actual system output, and $M(s)$ denotes the diagonal matrix $\frac{1}{s}$, with dimensions match the reference signal. This parameter accounts for the effect of the integrator present in the LQI

controller as thoroughly investigated in Section 3.3.1. Next, the transfer function with respect to Figure 3.3, which links the input signal u_a to r, will be derived. Here, r is the residual generated from Equation (61). To construct the residual, first, let us drive the y and $\hat{y}_o$ as follows:

$$y(s) = y_a(s) + G_{yu_a}(s)u_a(s) + G_{yr}(s)r(s) \tag{65}$$

$$\hat{y}_o(s) = G_{\hat{y_o}r}(s)r(s) + G_{\hat{y_o}y_a}(s)y_a(s) + G_{\hat{y_o}u_a}(s)u_a(s) \tag{66}$$

By considering Equations (65) & (66), the Equation (61) can be reconstructed as follows:

$$r(s) = \hat{y_o}(s) - y(s) \tag{67}$$

$$r(s) = (I - G_{\hat{y_o}y_a}(s))y_a(s) + (G_{yu_a}(s) - G_{\hat{y_o}y}(s)G_{yu_a}(s))u_a(s) + (G_{yr}(s) - G_{\hat{y_o}r}(s))r(s) \tag{68}$$

Equation (68) defines the transfer function from u_a to r. However, it includes a term for $y_a(s)$, which represents the covert attack formulation $y_a(s) = G_{y_au_a}(s)u_a(s)$. This type of attack uses complete system knowledge, including the transfer function, to inject the attack and remain undetectable. With this consideration Equation (68) is written as follows:

$$r(s) = (I - G_{\hat{y_o}y_a}(s))G_{y_au_a}(s)u_a(s) + (G_{yu_a}(s) - G_{\hat{y_o}y}(s)G_{yu_a}(s))u_a(s) + (G_{yr}(s) - G_{\hat{y_o}r}(s))r(s) \tag{69}$$

To drive the $G_{ru_a}(s)$ and apply the attack transfer function concerning the $G_{y_au_a}(s) = -G_{yu_a}(s)S_{i_c}u_a(s)$ is represents as follows:

$$\begin{aligned} G_{ru_a} &= G_{y_au_a}(s) - G_{\hat{y_o}y_a}(s)G_{y_au_a}(s) + G_{yu_a}(s) - G_{\hat{y_o}y}(s)G_{yu_a}(s) \\ &= -G_{yu_a}(s)S_{i_c} + G_{yu_a}(s) - G_{\hat{y_o}y_a}(s)G_{y_au_a}(s) - G_{\hat{y_o}y}(s)G_{yu_a}(s) \\ &= G_{yu_a}(s)(I - S_{i_c}) + G_{yy_a}(s)G_{yu_a}(s)S_{i_c} - G_{\hat{y_o}y}(s)G_{yu_a}(s) \end{aligned} \tag{70}$$

While $G_{\hat{y_o}y_a}(s) = G_{\hat{y_o}y}(s)$ and Equation (64), the above equation can be written as follows:

$$
\begin{aligned}
G_{ru_a}(s) &= -G_{yu_a}(s)(-I + S_{i_c}) + G_{\hat{y_o}y}(s)G_{yu_a}(s)(S_{i_c} - I) \\
= (G_{\hat{y_o}y}(s) - I)G_{yu_a}(s)(S_{i_c} - I) &= (G_{\hat{y_o}y}(s) - I)T(s)S_{i_c}^{-1}(S_{i_c} - I) \\
&= (G_{\hat{y_o}y}(s) - I)T(s) - (G_{\hat{y_o}y}(s) - I)T(s)S_{i_c}^{-1}
\end{aligned} \tag{71}
$$

The above equation describes the transfer function $G_{ru_a}(s)$, which relates the residual signal r to the input signal u_a in the presence of a Kalman filter and the effects of the LQI controller.

Regarding having an optimal coding matrix, the objective is defined in a way that maximizes the effect of the attack input u_a on the residual in the meanwhile required to minimize the effect of the attack input on the system by these considerations, the optimization objective function is defined as follows:

$$
\max_{S_{i_c}^{-1}} J_{S_{i_c}^{-1}} = \max_{S_{i_c}^{-1}} \frac{\|G_{ru_a}(s)\|_-}{\|G_{yu_a}(s)\|_\infty} \tag{72}
$$

As indicated by the proof in [7], their work demonstrates that the solution to this optimization objective in $s = j\omega_0$ is justified. The nature of the norm in the optimization problem solves the objective function over all frequencies. For non-square systems, it becomes essential to determine an appropriate right pseudo-inverse. The considered solution is shown as follows:

$$
\begin{aligned}
\max_{S_{i_c}^{-1}} J_{S_{i_c}^{-1}} &= \max_{S_{i_c}^{-1}} \frac{\underline{\sigma}((G_{\hat{y_o}y}(j\omega_0) - I)T(j\omega_0) - (G_{\hat{y_o}y}(j\omega_0) - I)T(j\omega_0)S_{i_c}^{-1})}{\bar{\sigma}(T(j\omega_0)S_{i_c}^{-1})} \\
&\leq \max_{S_{i_c}^{-1}} \frac{||(G_{\hat{y_o}y}(j\omega_0) - I)T(j\omega_0)||_- + ||(G_{\hat{y_o}y}((j\omega_0) - I)T(j\omega_0)S_{i_c}^{-1}||_-}{||T(j\omega_0)S_{i_c}^{-1}||_\infty}
\end{aligned} \tag{73}
$$

As shown in Equation (73), the required constraint is not to ruin the invertibility condition of the coding-decoding matrix. The condition $\underline{\sigma}(S_{i_c}^{-1}) \approx \bar{\sigma}(S_{i_c}^{-1}) \to 0$ implies that, as time increases, the singular values of $S_{i_c}^{-1}$ approach zero, causing the matrix to lose its ability to stretch or transform space in any significant way. This progression results in $S_{i_c}^{-1}$ becoming nearly singular, as both its smallest and largest singular values diminish towards zero, indicating a loss of invertibility. Consequently, $S_{i_c}^{-1}$ behaves increasingly like a non-invertible matrix, rendering it ineffective in preserving information through transformations; therefore, the optimization should be solved in the

specific boundary condition as follows:

$$\begin{aligned} \gamma &= \sigma_{max}(N(j\omega_0)) \\ \rho &= \sigma_{min}(T(j\omega_0)) \end{aligned} \tag{74}$$

Conditions (74) demonstrate boundary conditions in a steady state, which is applied on Equation (73) to get feasible sub-optimal solutions. Here, $N(j\omega_0) = (G_{\hat{y}_o y}(j\omega_0) - I)T(j\omega_0)$, and this term has the most effect on the designed coding-decoding process, so the simplified constant coding matrix concerning the constraints is designed as follows:

$$S_{i_c}^{-1} = \frac{\gamma}{\rho} N^{-1} \tag{75}$$

The coefficient $\frac{\gamma}{\rho}$ is termed sub-optimal because it provides a stable approximation rather than the best possible performance across all conditions.

3.4.2 Periodic Coding Matrix

As illustrated in Equation (75), the constant coding matrix is represented as a single matrix to enhance the detection process concerning stealthy covert attacks. However, the single matrix is not complex enough to remain undetectable during the mission from the attacker side, so this constant coding matrix requires to be more complex to address the security concern as well. This salvation is considered at $\omega_o = 0$ as this attack is applicable to the steady-state condition. Regarding the security concern, the Singular Value Decomposition (SVD) technique is applied to a part of $G_{ry}(j0)$, which proves to be more effective for attack detection problems. This focus involves no loss of generality, as $G_{ry}(j0)$ can be considered a part of $N^{-1}(j0)$ because the $\hat{y}$ enhances the detection functionality. Let us concern alternative representation for the $G_{ry}(j0)$ as a part of $N(j0)$ as follows:

$$\begin{aligned} G_{ry}(j0) &= \begin{bmatrix} I & \tilde{C}(sI - (\hat{A} - L\hat{C})) \end{bmatrix} \begin{bmatrix} -I \\ L_1 - \hat{B} \end{bmatrix} \\ &= \begin{bmatrix} I & \tilde{C}(sI - (\hat{A} - L\hat{C})) \end{bmatrix} \begin{bmatrix} \beta_s \end{bmatrix} \end{aligned} \tag{76}$$

Equation (76), the matrix $\begin{bmatrix} -I \\ L_1 - \hat{B} \end{bmatrix}$ predominantly influences both the attack input and its impact on the residual. L_1 is regarded as the impact of the KF gain on the affected output, as described in Equation (55). By applying the Singular Value Decomposition (SVD) technique, the largest singular value of this matrix can be used to replace the entirety of $G_{ry}(j0)$, generating various possible matrices with critical properties relevant to attack detection. Furthermore, strategically positioning this vector within the existing matrix structure offers multiple configurations of coding matrices, each retaining the essential characteristics required for effective attack detection as follows:

$$S_i^{-1} = \frac{\gamma}{\rho}[(s_i - I)T(j0)]^{-1}, i = 1, 2, ..., m! \tag{77}$$

Equation (77) demonstrates the periodic matrix that can be generated $m!$ distinct coding matrices, where the value m corresponds to the order of the B-matrix within the system. Each of these matrices is designed with a unique structure, requiring a specific index. Here, s_i is one of the column permutations of the matrix V^H, as shown as follows:

$$\begin{bmatrix} -I \\ L_1 - \hat{B} \end{bmatrix} = U \cdot \Sigma \cdot V^H \tag{78}$$

Equation (78) represents the SVD technique applied on $\begin{bmatrix} -I \\ L_1 - \hat{B} \end{bmatrix}$ matrix to extract the most output direction matrix that used as a part of Equation (77) to design s_i.

The configuration of S_i is strategically chosen to align the dimensions of the reference R with that of the coding matrix. Specifically, if the target input is represented by a vector containing four elements, the corresponding S_i is structured as a 4×4 matrix, ensuring the designed matrix is suitably dimensioned to interact with every input vector element. Although the discussed design process offered a variety of advantages, such as the capability of detecting covert attacks, but also remaining complex enough to be inaccessible from the attacker's perspective, due to the research focusing on an application-based subject, the concept must apply to the designed application. Within the context of the CPS framework, two critical considerations must be addressed to ensure the

solution's applicability. First, storing the designed matrices in the plant in a manner that allows for timely usage is challenging. This requires capturing all coding possibilities and storing them as a dictionary on the plant side before the mission's commencement. It is assumed that the attacker cannot access the existing dictionary within the plant. Second, the synchronization of coding and decoding processes must be ensured to fulfill the detection objective. As previously mentioned, this involves establishing a secure channel to securely transmit the chosen coding matrix index to the plant side for decoding.

3.4.3 Dictionary Design:

As discussed in the previous section, addressing the challenge of enabling the system to decode the strategically designed coding matrix requires revising the coding approach and setting a pre-defined dictionary. This dictionary should be compiled before the plant launches its mission, and the designed dictionary will include all possible combinations of the coding matrix. It should be securely stored within the plant and C&C side. This measure makes the decoding process feasible. The existence of this dictionary at both the C&C and on the plant side highlights the critical role of a secure communication channel dedicated just to transmitting the **index** corresponding to the coding matrix from the C&C to the plant. This protocol allows the plant to accurately identify the coding matrix and execute the decoding process, thus effectively safeguarding against sophisticated stealth attacks. This section outlines the strategy behind creating the dictionary and transmitting the specific coding matrix identifier via the secure communication channel. The following algorithm succinctly outlines the entire sequence of steps related to establishing the dictionary and selecting the coding matrix:

Algorithm 1 Dictionary Design & Data Transmission via Secure Channel

1: Calculate all the unique possibilities for $S_i \in R^{m\times m}$, $j \leftarrow m!$ from Equation (77).
2: $S_{\text{large}} \in R^{(jm\times m)}$ is structured.
3: $index \leftarrow$ Randomly select an index from 1 to j.
This step involves randomly selecting S_i from S_{large}.
4: Selected index is transmitted through the secure channel to the plant side.
5: The plant, upon receiving the index, selects the corresponding S_i from the existing dictionary of S_{large} for the decoding process.

The dictionary matrix S_{large} design process involves the following key steps:

(1) **Define the dictionary size**: The dictionary size is linked to the configuration and original size of the coding matrix, and all the distinct combinations of this matrix can be assumed. For instance, if the dimensions of the S_i is 4×4, the total number of unique representations of this matrix, considering both rows and columns, would amount to $4!$ scenarios. This $4!$ figure represents all distinct potential configurations for the coding matrix, effectively defining the dictionary's capacity in this design context. Given these definitions, the size of S_{large} can be expressed as:

$$S_{\text{large}} \in R^{(jm \times m)}$$

where $m \times m$ is the dimensions of S_i,and j is the number of all possibilities of the S_i in that specific scenario. It is necessary to mention that the designed dictionary of this part is the only thing shared between the plant and the C&C before starting the mission.

(2) **Random S_i selection from S_{large}** : Once the coding matrix is designed and the dictionary is established, it becomes essential to outline the process of selecting the appropriate coding matrix from the dictionary during the mission. This step is processed completely randomly and could even be repetitive. As mentioned previously, for the considered case, this random selection is based on random permutations, which are applicable for symmetric groups throughout the simulation. The only information the C&C picks from this selection is the matrix's index within the dictionary and sends it to the plant side through the communication channel for the decoding process.

The designed procedures represent a coding-decoding process capable of detecting covert attacks while enhancing the detector's accuracy in C&C. By designing multiple coding matrices, this approach complicates an attacker's ability to determine the specific coding matrix in use, thereby increasing system security and providing the defender with additional time for mitigation and compensation. The proposed design also effectively detects replay attacks. This concept will be explored in detail in the following sections.

Remark of proposed Detection Procedure Counter Replay Attack

As mentioned in Figure 1.6, a replay attack requires resources for disclosure and disruption to pose this attack and remain undetectable. In contrast, a covert attack not only demands similar access to these resources but also necessitates a system knowledge of the system, as shown in the figure, to remain undetected, executing its influence stealthily. This distinction implies that a replay attack can be viewed as a subset of a covert attack due to its less complex requirements. The proposed solution in this research is designed to detect covert and replay attacks effectively. Although this method requires more resources compared to techniques specifically tailored to counter replay attacks alone, it provides the added advantage of detecting both attacks, as validated by the simulation results presented in the following section.

3.4.4 Overview

As a brief overview before delving into the simulation results, it is essential to highlight key components of the designed procedure to ensure complete clarity. Section 3.4 investigated the entire design procedure that aids traditional chi-square detection. As represented in Equation (74), the applied coding decoding technique can potentially raise false alarms, as indicated in Equation (63), and detect covert and replay attacks in the C&C. This demonstrates the practicality of the designed matrix.

However, other critical steps were also examined in this section. For instance, to address the security issue where an attacker could not easily identify the designed coding procedure, in Equation (77), a periodic coding matrix was designed to address the security concern. This design enhances the complexity of the coding, making it more difficult for attackers to decipher. Additionally, the synchronization problem of the coding/decoding process was addressed in the dictionary design as discussed in Section 3.4.3. Also, the secure channel is considered to transfer the chosen coding label from C&C to the plant side; this label transferring also ensures more security rather than sending the entire coding through the communication channel and saves more resources. These essential issues were tackled with an application-based research perspective, ensuring practical applicability in real-world scenarios. With these clarifications, the following section presents the simulation results applied to the 6DOF quadrotor, demonstrating the validity of the research at this stage.

3.5 Simulation Results

This part investigates the simulation results, focusing on evaluating the performance of a linearized quadrotor within a CPS configuration, particularly under stealthy cyber-attacks such as covert or replay attacks. These scenarios are essential to assessing the detector's ability to identify attacks typically undetectable by standard detection mechanisms, highlighting the advanced nature of the threats facing CPS today.

Sub-section 3.5.1 thoroughly outlines the parameters, inputs, outputs, and references, detailing the relationships among them to be examined in this research. This section provides a clear foundation for the research by explicitly defining the variables involved. The sub-section 3.5.2 shows the operation details of the case study under normal, healthy conditions for two different scenarios with different references from C&C, providing practical evidence of the effectiveness of the examined UAV. The exploration progresses in the Sub-section 3.5.3 for two mentioned scenarios under covert attack signals. It visually compares the outputs observed from the plant side and the outputs received by the C&C for each scenario.

Following this sub-section, 3.5.4 the inefficacy of the Chi-square detector is critically examined across the defined attack scenarios. This section unveils its limitations in identifying the sophisticated covert attacks presented, challenging the detector's recognition capabilities within these extreme conditions. Sub-section 3.5.5 presents the design and implementation of a coding scheme to enhance the Chi-square detector's ability to counteract covert attacks. The effectiveness of this coding scheme is validated through its application, with outcomes displayed to affirm its utility. Including coding matrices in this evaluation ensures an assessment, establishing the coding scheme's role in enhancing the Chi-square detector's performance against covert attacks. Finally, the last subsection validates the coding-decoding design to detect the replay attack, as mentioned in the remark of the previous section.

3.5.1 Specifying System Parameters

First, let us consider the following parameters as the state matrices of a system from [77].

$$A = \begin{bmatrix} 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 \\ 0 & 0 & g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 \\ -g & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \end{bmatrix}, B = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & \frac{l}{I_{xx}} & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & \frac{l}{I_{yy}} & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & \frac{1}{I_{zz}} \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ \frac{1}{m} & 0 & 0 & 0 \end{pmatrix}, C = \begin{pmatrix} 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 1 & 0 \end{pmatrix}, D = \begin{pmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{pmatrix}$$

As shown in Equation (20), the matrices A, B, C, and D constitute the core components of the state-space model that characterizes the system's behaviour. These parameters represent the quadrotor's physical attributes and play a crucial role in flight behaviour. As explicitly explained in Section 2.2 as a background of the quadrotor. The system's dynamics are characterized by moments of inertia I_{xx} and I_{yy} at $7.5 \times 10^{-3}\,\text{kg} \cdot \text{m}^2$ for the x and y axes, and a larger I_{zz} of $1.3 \times 10^{-2}\,\text{kg} \cdot \text{m}^2$ for the z-axis, indicating asymmetrical mass distribution. The arm length l is set to $0.23\,\text{m}$, critical for the system's torque dynamics. Gravitational acceleration is considered as $9.81\,\text{m/s}^2$, the standard on Earth. The system's mass is $0.65\,\text{kg}$. These parameters are integral to developing and tuning control algorithms, ensuring the quadrotor's stability, maneuverability, and overall flight performance.

To clarify, essential parameters mentioned for the dynamics and control of a quadrotor, characterized by its four rotors, are shown as the following table :

Symbol	Value
I_{xx}	$7.5 \times 10^{-3}\,\text{kg} \cdot \text{m}^2$
I_{yy}	$7.5 \times 10^{-3}\,\text{kg} \cdot \text{m}^2$
I_{zz}	$1.3 \times 10^{-2}\,\text{kg} \cdot \text{m}^2$
l	$0.23\,\text{m}$
g	$9.81\,\text{m/s}^2$
m	$0.65\,\text{kg}$

Table 3.1: Quadrotor parameters.

Here, the focus is on a quadrotor, and the state vector X contains essential attributes such as spatial

positions, angular orientations, and their corresponding velocities, included in the state vector X as shown following:

$$X = \begin{bmatrix} \phi & \dot{\phi} & \theta & \dot{\theta} & \psi & \dot{\psi} & y & \dot{y} & z & \dot{z} & x & \dot{x} \end{bmatrix}^T \tag{79}$$

The parameters represent the state variables of a quadrotor in 3D space: ϕ and $\dot{\phi}$, θ and $\dot{\theta}$, ψ and $\dot{\psi}$ for the roll, pitch and yaw angles and their rates respectively, z and $\dot{z}$ for the vertical position and velocity, and x, $\dot{x}$, and y, $\dot{y}$ for the horizontal positions and velocities in the x and y directions, respectively. Also, the relation between the quadrotor states is represented as follows:

$$\begin{aligned} x_1 &= \phi & x_7 &= y \\ x_2 &= \dot{x}_1 = \dot{\phi} & x_8 &= \dot{x}_7 = \dot{y} \\ x_3 &= \theta & x_9 &= z \\ x_4 &= \dot{x}_3 = \dot{\theta} & x_{10} &= \dot{x}_9 = \dot{z} \\ x_5 &= \psi & x_{11} &= x \\ x_6 &= \dot{x}_5 = \dot{\psi} & x_{12} &= \dot{x}_{11} = \dot{x} \end{aligned}$$

The state-space representation is crucial to this research in demonstrating the relation between the states and the effect of the input on the outputs. As shown in Section 3.3, the main objective in this case is the set-point regulation problem. This means with respect to the designed LQI controller, the input reference signal is augmented with an integral part of the controller that actively tries to minimize the difference between the reference signal and the outputs. Here, the reference signal is a vector with four elements, as shown as follows:

$$R(t) = \begin{bmatrix} r_1 & r_2 & r_3 & r_4 \end{bmatrix} \tag{80}$$

According to the described objectives, some of the outputs are intended to align with this reference signal. The considered outputs in this research are shown as follows:

$$y(t) = \begin{bmatrix} \phi & \theta & \psi & y & z & x \end{bmatrix} \tag{81}$$

As described in Equation (79), this research evaluates the positions of the quadrotor to fulfill the set-point regulation objective, the reference signal has a direct effect on the $\begin{bmatrix} \psi & y & z & x \end{bmatrix}$ outputs. This information keeps the research consideration close to reality.

3.5.2 System under Healthy Conditions:

This section considers two scenarios to evaluate the efficiency of the simulation results. These scenarios investigate the set point regulation problem through a detailed analysis of the simulation outcomes.

Scenario A: In this scenario, the reference signal is $R(t) = \begin{bmatrix} 30 & 2 & -5 & 8 \end{bmatrix}$. The observed outputs from C&C with respect to the defined reference are demonstrated as follows:

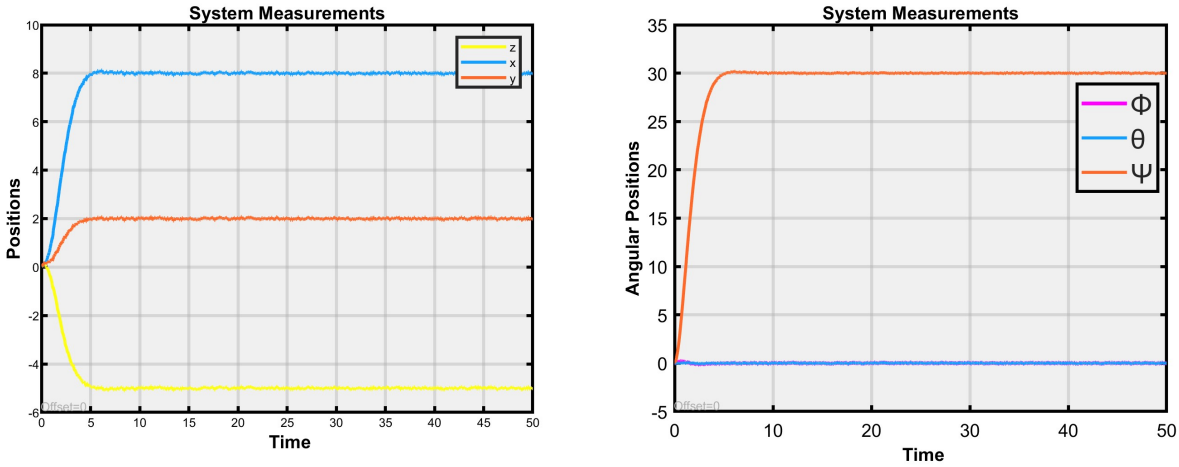

Figure 3.4: Sensor measurements for scenario A.

Figure 3.4 demonstrates the system outputs (y, z, x) over time which follow the predefined reference. The position outputs were brought closer to the predefined reference signal, effectively addressing the regulation problem. Moreover, illustrates the system's angular positions over time in response to the defined reference signal. It displays three lines representing the angular positions (ϕ, θ, ψ),

each corresponding to a distinct rotational axis. The figure demonstrates the effect of the predefined reference only on one of the angular positions which is ψ.

Scenario B: In this scenario, the reference signal is $R(t) = \begin{bmatrix} 45 & 6 & 10 & -5 \end{bmatrix}$. The outputs with respect to the defined reference are demonstrated as follows:

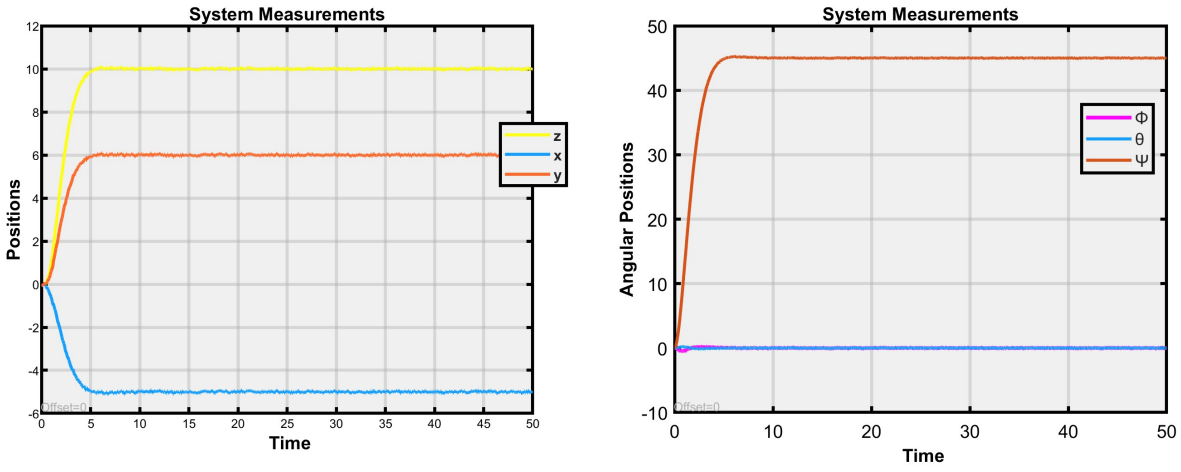

Figure 3.5: Sensor measurements for scenario B.

Figure 3.5 illustrates the system outputs (y, z, x) over time, demonstrating their alignment with the predefined reference for scenario B. The position outputs were adjusted to closely match the predefined reference signal, effectively resolving the regulation problem. Figure 3.5 displays the angular positions of the system over time for the healthy scenario B that is specified with the predefined reference signal. The plot includes three lines, each representing an angular position (ϕ, θ, ψ) associated with a different rotational axis. This figure emphasizes that the predefined reference signal primarily influences the angular position ψ.

3.5.3 Analyzing System Vulnerability under Covert Attack:

This section investigates the stealthy nature of covert attacks and demonstrates the system outputs from both the plant side, which is inaccessible and the C&C side. As discussed mathematically in Section 3.2, the covert attack intelligently cancels its effect within the output communication channel. It prevents the C&C from recognizing that the plant side is in significant danger. The previous two scenarios with different attack input signals are considered to evaluate the system outputs

under covert attack conditions in this stage of research.

Scenario A: In the initially examined scenario, a covert attack impacts the system starting at the 40th time unit, and $u_a = \begin{bmatrix} 30 & 2 & -3 & 4 \end{bmatrix}$. However, the reference signal is the same as scenario A of the last subsection. As a nature of the covert attack discussed in Section 3.2, this kind of attack has access to the system knowledge. It can eliminate the effect of input on the system in the output communication channel. The comparison of the system's actual outputs from the plan side and the C&C in the first scenario is shown as follows :

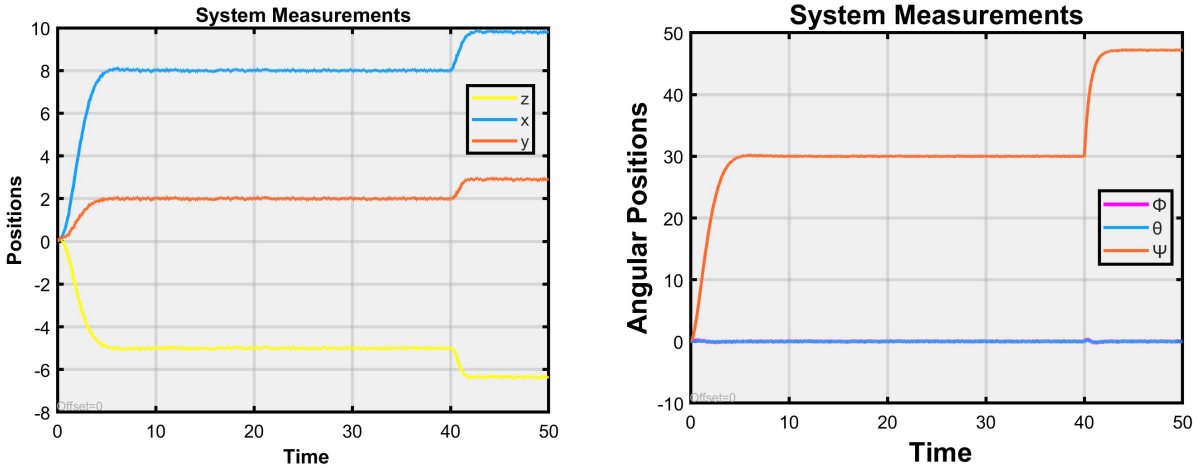

Figure 3.6: Plant side outputs under the covert attack.

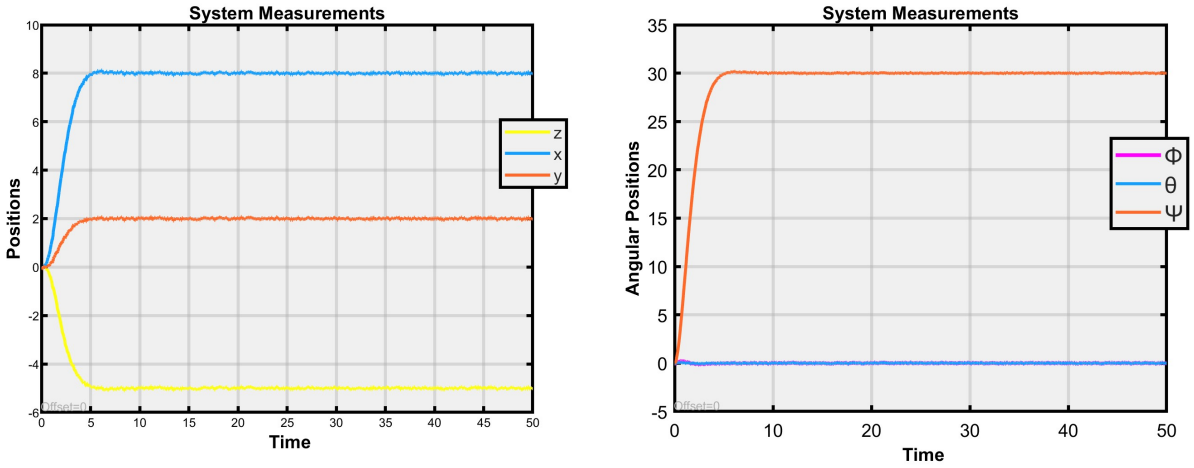

Figure 3.7: C&C side outputs under the covert attack.

As mentioned for the scenario A, the expected healthy set point that the UAV should follow is $r(t) = \begin{bmatrix} 45 & 6 & 10 & -8 \end{bmatrix}$. However, starting from time 40, the UAV is misled and follows a new trajectory due to the attacker's injection of a new input. This malicious reference harms the system, but since the considered attack is intelligent, this manipulation remains undetectable from the C&C side. Consequently, the attack remains hidden from the C&C center.

Scenario B: In the second examined scenario, a covert attack impacts the system starting at the 30th time unit, and $u_a = \begin{bmatrix} 10 & 6 & 5 & -4 \end{bmatrix}$. The comparison of the system's actual outputs from the plan side and the C&C in the second scenario is shown as follows:

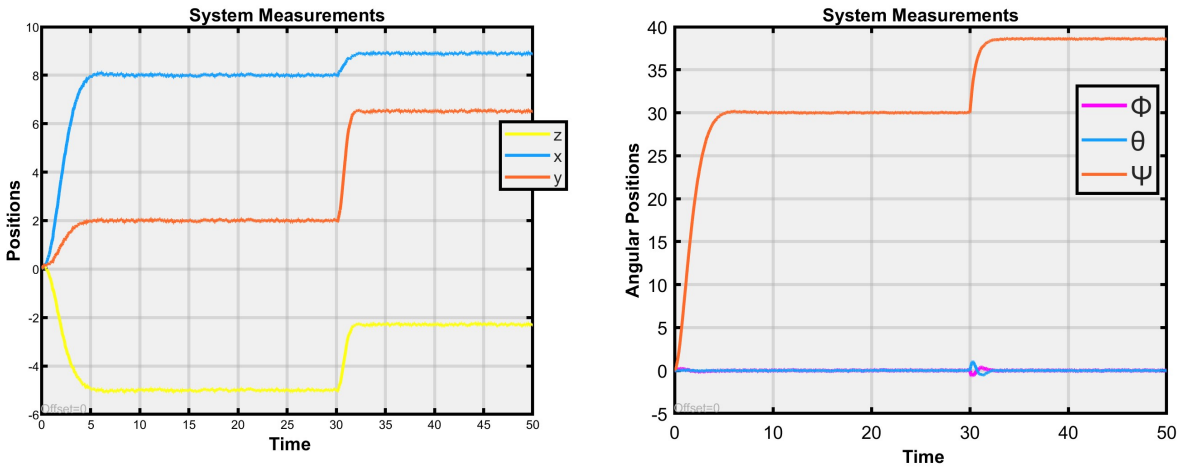

Figure 3.8: Plant side outputs under the covert attack.

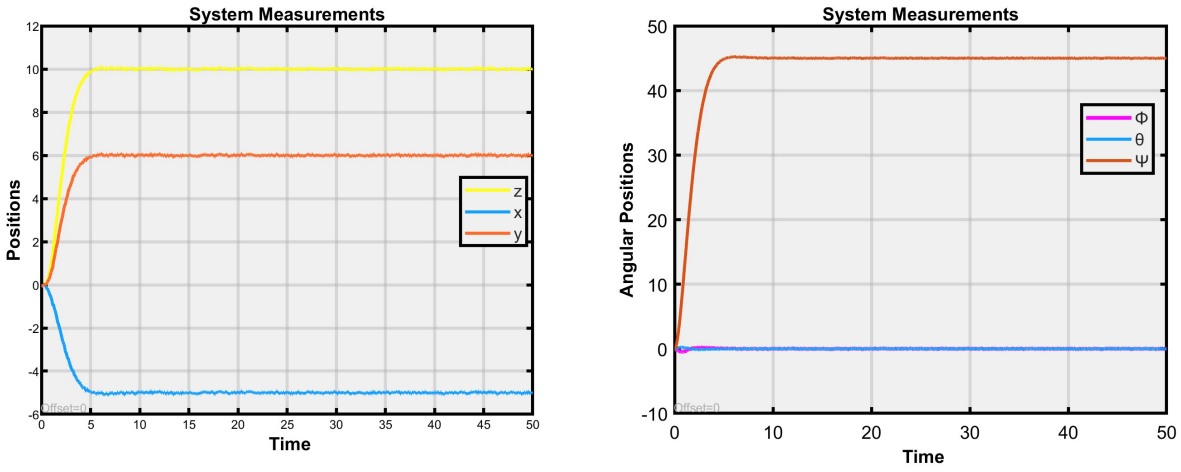

Figure 3.9: C&C side outputs under the covert attack.

Here, the attack signal is added to the existing input reference signal of scenario B from the previous subsection. The additive effect is subsequently cancelled out at the plant side output, ensuring the stealthy objective of the attack is achieved by the attacker.

As demonstrated in this section, the attack remains completely undetectable from the C&C side. In the subsequent section, one of the most powerful traditional or passive detection methods, known as the χ^2 test, is implemented to ensure that for each scenario, the C&C is unable to detect the attack using this existing traditional detection strategy.

3.5.4 χ^2 Detector's Effectiveness against Covert Attack:

As mathematically described in Section 3.3.3, this powerful detection strategy has two main steps: defining the residual and establishing the detection threshold using the Monte Carlo method. The residual, $r(t)$, is the difference between the actual system output and the estimated output from the Kalman filter, as noted in Equation (55). The detection threshold is determined by running the simulation 100 times under healthy conditions with noise effects. The threshold is set to 2 for this specific UAV by considering the sensor and the process noise, ensuring optimal detection accuracy. The Chi-square detection approach analyzes the residuals and deviations between the system's actual outputs and the estimated outputs, providing a highly accurate statistical threshold to differentiate between typical variations and potential attack signals.

This section presents the Chi-square test results for the aforementioned scenarios, demonstrating the limitations of this highly effective detector, which can not accurately detect attacks.

These two figures demonstrate the χ^2 test results for both scenarios as follow:

Scenario A:

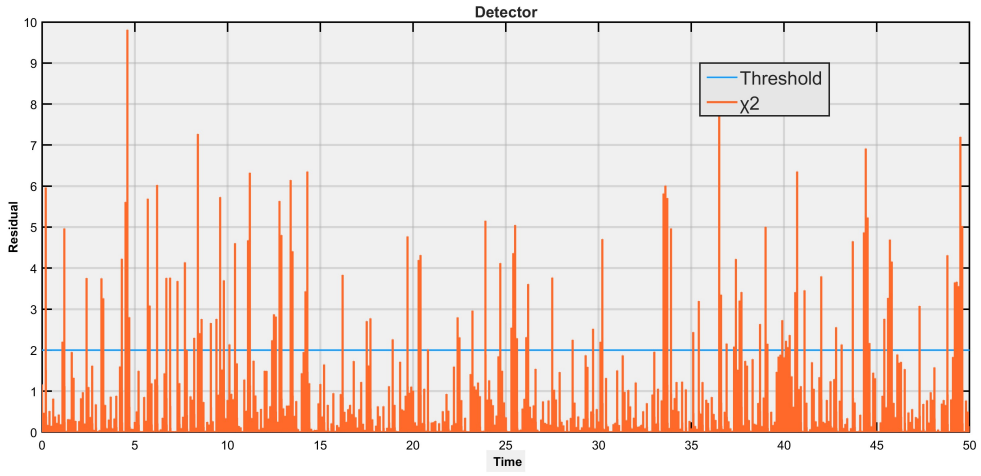

Figure 3.10: χ^2 test for scenario A.

Scenario B:

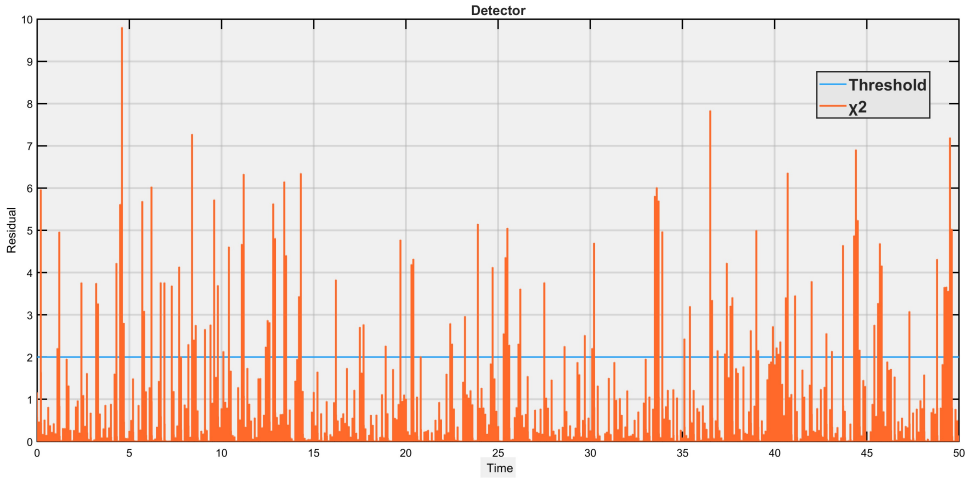

Figure 3.11: χ^2 test for scenario B.

As illustrated in these figures for both scenarios, there are instances where the χ^2 test exceeds the threshold; since it does not consistently remain above the threshold, these instances cannot be considered as detection in C&C center.

Although the Chi-square test is widely regarded as one of the most powerful detection strategies in the literature, it is not sufficiently capable of countering covert attacks. Therefore, this strategy requires enhancements to effectively detect covert attacks.

3.5.5 Coding-Decoding Effects:

As mathematically investigated in Section 3.4, the coding and decoding design is a procedure that can significantly enhance the effectiveness of the Chi-square test in detecting stealthy attacks. The decoding process is essential to neutralize the harmful effects caused by the coding within the system. As discussed in Section 3.4.2, the part of the periodic coding matrix is derived from the SVD of $\begin{bmatrix} -I \\ L_1 - \hat{B} \end{bmatrix}$ in the Equation (77) which gives the availability to generate different coding matrices instead of one. This approach effectively addresses the complexity objective by constructing the periodic coding matrix and aids in the security problem as mathematically discussed in Equation (77). The matrix S_i is regarded as each constant coding matrix, as depicted in Algorithm 1. This coding method in this case study can represent $4!$ matrices by rotating the V^H elements as described in (77). Consequently, the dictionary comprises 24 matrices with 4×4 elements. These matrices are arranged vertically, with each matrix labelled by its corresponding number. For instance, matrix five is located from columns 17 to 20 of the large coding matrix S_{large} as noted in the dictionary in the algorithm. The part of dictionary with respect to the evaluating case study is shown as follows:

$$S_1 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 5.2464 \times 10^{-16} & 0 & -0.1844 & -0.9828 \\ 3.5734 \times 10^{-15} & 0 & 0.9828 & -0.1844 \\ 1 & 0 & -3.3203 \times 10^{-15} & 1.2941 \times 10^{-15} \end{bmatrix} \quad S_2 = \begin{bmatrix} 0 & 1 & 0 & 0 \\ 5.2464 \times 10^{-16} & 0 & -0.9828 & -0.1844 \\ 3.5734 \times 10^{-15} & 0 & -0.1844 & 0.9828 \\ 1 & 0 & 1.2941 \times 10^{-15} & -3.3203 \times 10^{-15} \end{bmatrix}$$

$$S_3 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 5.2464 \times 10^{-16} & -0.1844 & 0 & -0.9828 \\ 3.5734 \times 10^{-15} & 0.9828 & 0 & -0.1844 \\ 1 & -3.3203 \times 10^{-15} & 0 & 1.2941 \times 10^{-15} \end{bmatrix} \quad S_4 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 5.2464 \times 10^{-16} & -0.1844 & -0.9828 & 0 \\ 3.5734 \times 10^{-15} & 0.9828 & -0.1844 & 0 \\ 1 & -3.3203 \times 10^{-15} & 1.2941 \times 10^{-15} & 0 \end{bmatrix}$$

$$S_5 = \begin{bmatrix} 0 & 0 & 1 & 0 \\ 5.2464 \times 10^{-16} & -0.9828 & 0 & -0.1844 \\ 3.5734 \times 10^{-15} & -0.1844 & 0 & 0.9828 \\ 1 & 1.2941 \times 10^{-15} & 0 & -3.3203 \times 10^{-15} \end{bmatrix} \quad S_6 = \begin{bmatrix} 0 & 0 & 0 & 1 \\ 5.2464 \times 10^{-16} & -0.9828 & -0.1844 & 0 \\ 3.5734 \times 10^{-15} & -0.1844 & 0.9828 & 0 \\ 1 & 1.2941 \times 10^{-15} & -3.3203 \times 10^{-15} & 0 \end{bmatrix}$$

$$S_7 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 5.2464 \times 10^{-16} & -0.1844 & -0.9828 \\ 0 & 3.5734 \times 10^{-15} & 0.9828 & -0.1844 \\ 0 & 1 & -3.3203 \times 10^{-15} & 1.2941 \times 10^{-15} \end{bmatrix} \quad S_8 = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 5.2464 \times 10^{-16} & -0.9828 & -0.1844 \\ 0 & 3.5734 \times 10^{-15} & -0.1844 & 0.9828 \\ 0 & 1 & 1.2941 \times 10^{-15} & -3.3203 \times 10^{-15} \end{bmatrix}$$

The eight matrices shown here represent the initial entries in a predefined dictionary, denoted as S_{large}, which consists of 24 matrices in total. This dictionary, S_{large}, has an order of 96×4 and serves as a foundational component in the mathematical framework of this research. Each submatrix, labelled S_i, is indexed by i, representing its unique position within the dictionary. The randperm function generates a random permutation of the integers 1 to n. As the randperm invokes the rand function, it consequently alters the seed value of the rand. Alternatively, another permutation may be produced. At each clock pulse(time interval), the index of the selected matrix is transmitted to the plant side via a secure channel. This coding-decoding process directly impacts the system outputs. However, without a stealthy attack, this process neutralizes its effects and does not harm the system. The following sections demonstrate the comparison of system outputs in the presence of the designed matrix for both attack and non-attack cases across two existence scenarios.

Scenario A:

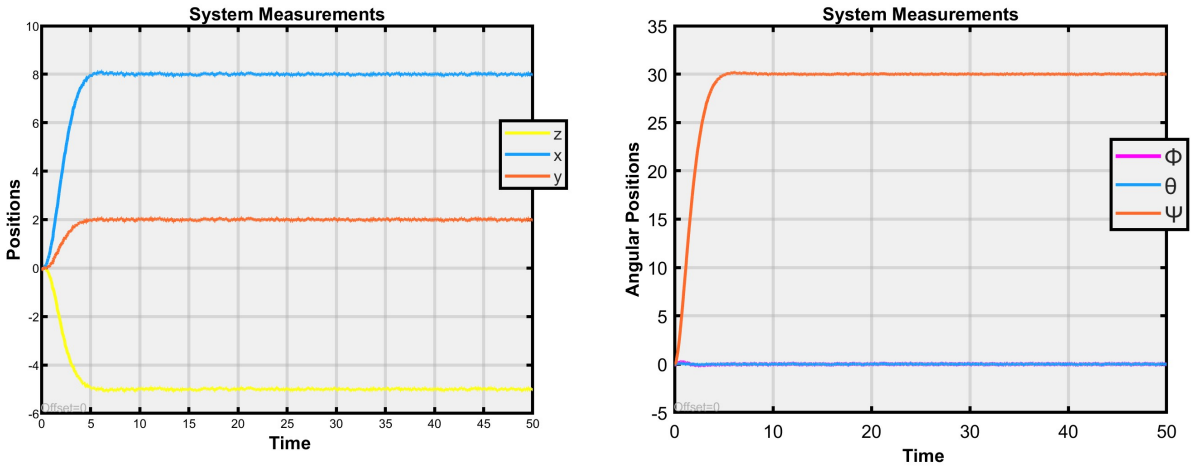

Figure 3.12: C&C side outputs without attack and coding-decoding effect.

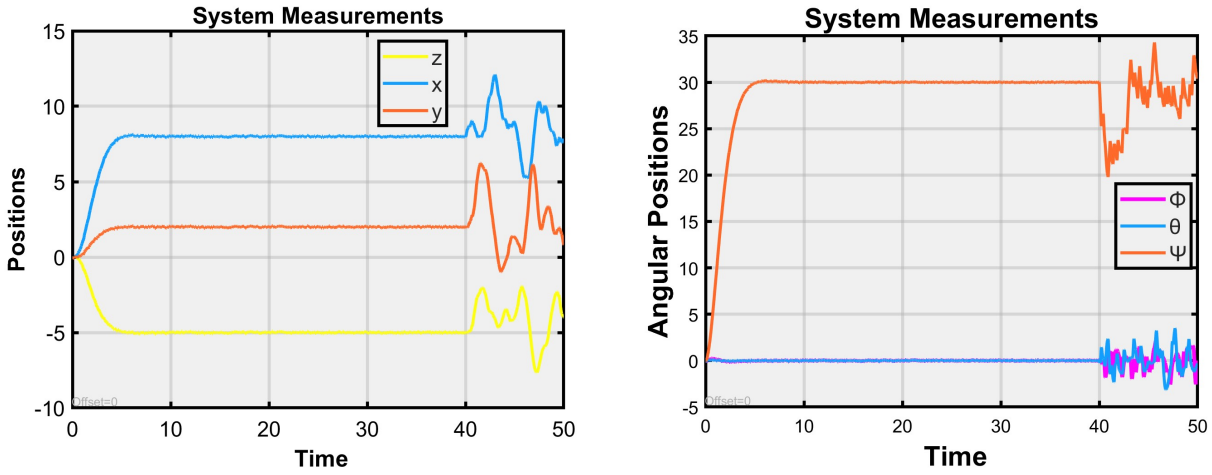

Figure 3.13: C&C side outputs with attack and coding-decoding effect.

Figure 3.12 illustrates the UAV's outputs from the C&C center, evaluating the synchronization of the coding-decoding design. This figure demonstrates that, under healthy conditions, the system does not perceive any adverse effects from the design. Conversely, Figure 3.13 presents the outputs from the C&C center in the presence of a covert attack. These figures clarify the impact of the coding-decoding design when a stealthy covert attack is occurring. Despite the attacker's attempts to mask the effects on the output, the impact cannot be removed from the decoding part, allowing the attack to be monitored at the C&C center.

Scenario B:

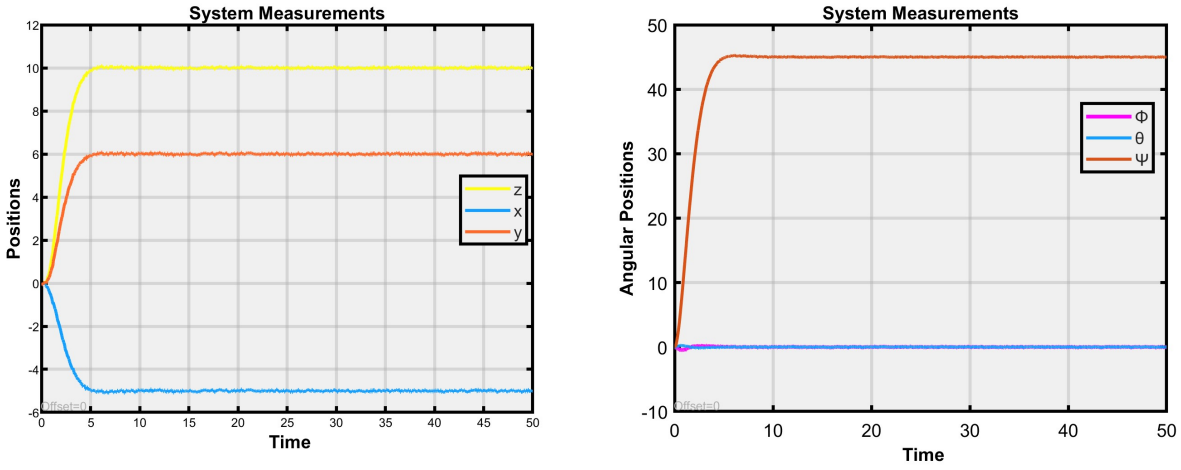

Figure 3.14: C&C side outputs without attack and coding-decoding effect.

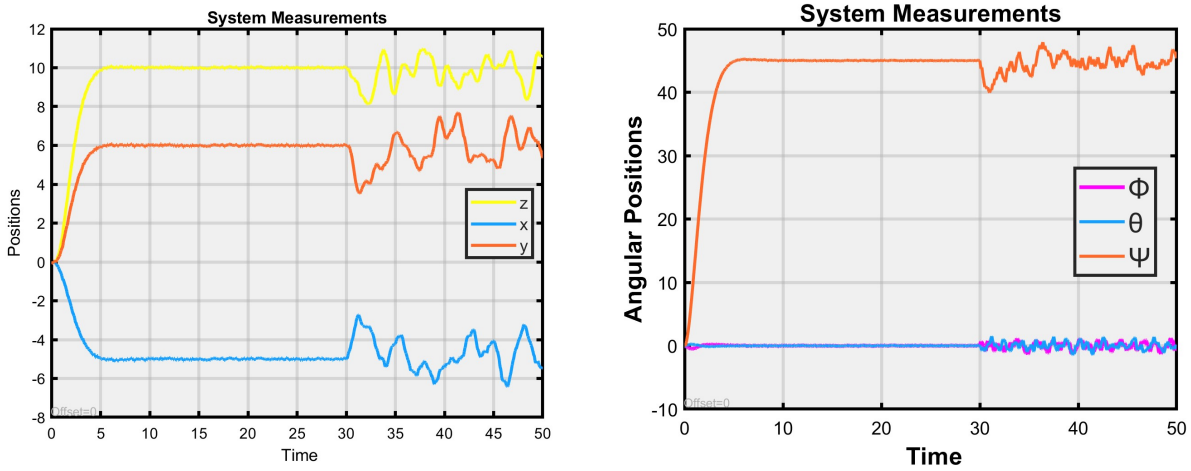

Figure 3.15: C&C side outputs with attack and coding-decoding effect.

Figure 3.14 showcases the UAV's outputs from the C&C center, assessing the synchronization of the coding-decoding design. This figure indicates that, in a healthy scenario, the system experiences no negative effects from the design. In contrast, Figure 3.15 depicts the outputs from the C&C center during a covert attack for the second scenario, which is accurate at time 30. These figures illustrate the influence of the coding-decoding design under a stealthy covert attack. Even though the attacker attempts to conceal the effects on the output, the impact persists in the decoding part, enabling the C&C center to monitor the attack.

3.5.6 Impact of Coding Design on χ^2 Detection:

The coding-decoding design can effectively reveal the presence of covert attacks in the output, allowing for easy monitoring by the C&C component. The decision-making unit is responsible for raising a red flag alarm based on the evaluation and comparison with a predefined threshold, which has been established using the Monte Carlo method described in Section 3.3.3. The following figure illustrates the output of the chi-square detector for both scenarios, showing the impact of the coding process on the data. The result of the chi-square detector for Scenario A in the presence of the coding design is demonstrated as follows:

Scenario A:

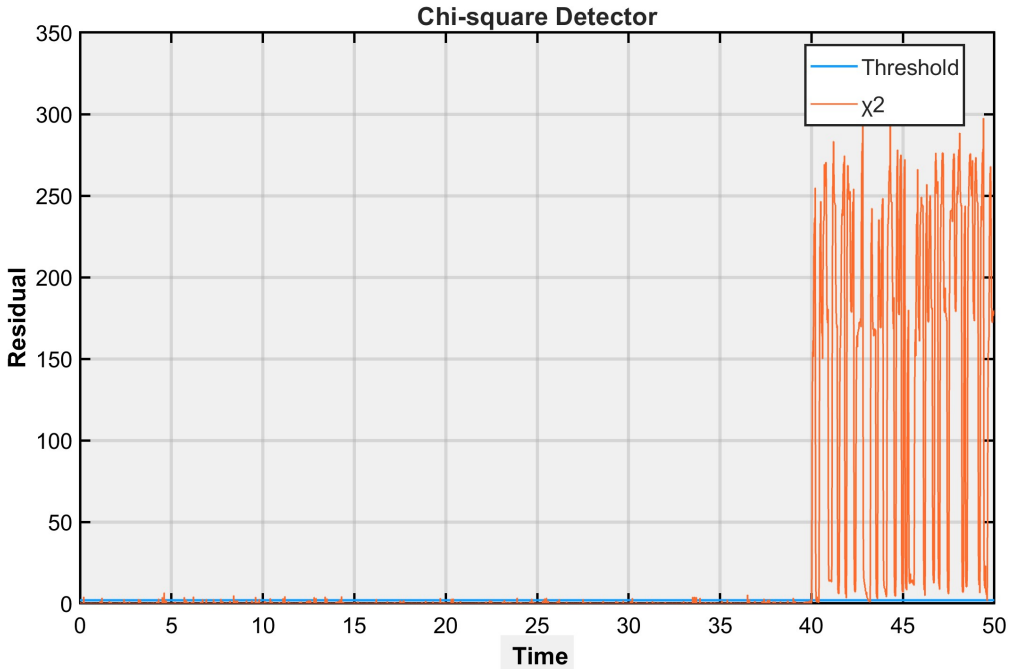

Figure 3.16: χ^2 test for scenario A.

Figure 3.16 depicts the χ^2 output, which compares the residual with the predefined threshold. According to the nature of the covert attack, this attack should remain undetectable. However, due to the effect of the decoding matrix, the attack becomes detectable from time 45, which marks the beginning of the covert attack's occurrence. Fortunately, this design does not affect the system output before the attack's occurrence. The following figure illustrates the χ^2 output for scenario B, in the presence of both the attack and the coding design, as follows:

Scenario B:

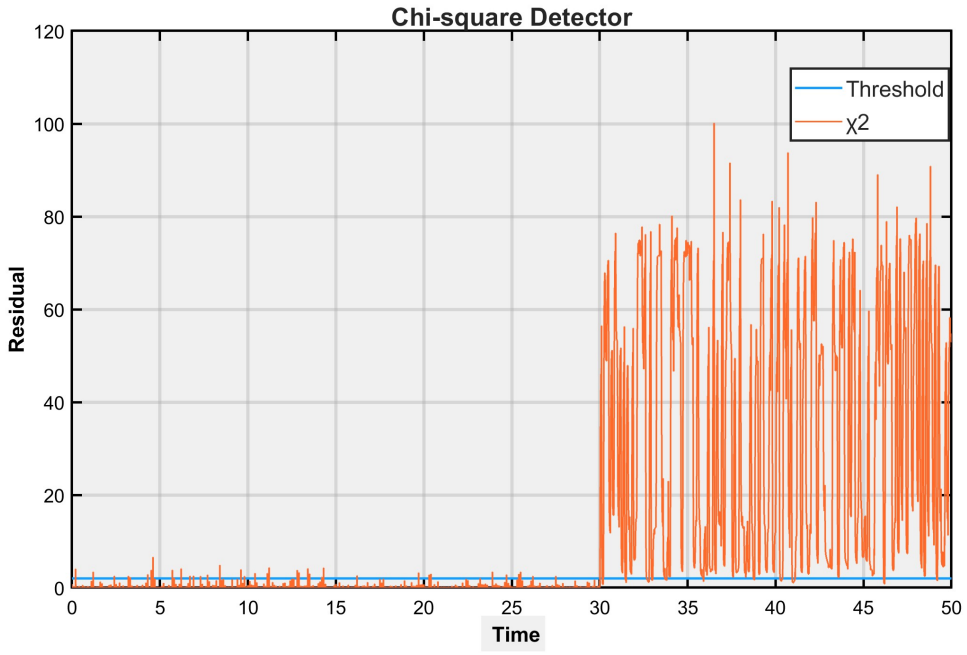

Figure 3.17: χ^2 test for scenario B.

Figure 3.17 presents the χ^2 output, which evaluates the residual against a predefined threshold. In theory, a covert attack should remain hidden. However, the decoding matrix reveals the attack starting at time 30, indicating the commencement of the covert activity. It is reassuring to note that this design does not alter the system output before the attack's initiation.

These figures illustrate the impact of the decoding process on the attack input, as mathematically demonstrated in Section 3.4.1. This section highlights the enhancement of χ^2 detection, which is augmented by the coding-decoding design. Although the mathematical representation is not only considered for detection, this design can be considered for future steps to mitigate the effects of the existing attack.

3.5.7 Evaluation of Proposed Detection Procedure Counter Replay Attack

This subsection provides a practical evaluation of the proposed solution, specifically assessing the effectiveness of the coding-decoding approach in detecting replay attacks as well, as highlighted in the methodological remark. In this case study, the UAV model with 6DoF remains the subject of analysis; however, it is now considered vulnerable solely to replay attacks. Initially, the performance of the UAV is assessed under two conditions: a health scenario and one involving a replay attack without the presence of coding-decoding as follows:

Replay attack scenario:

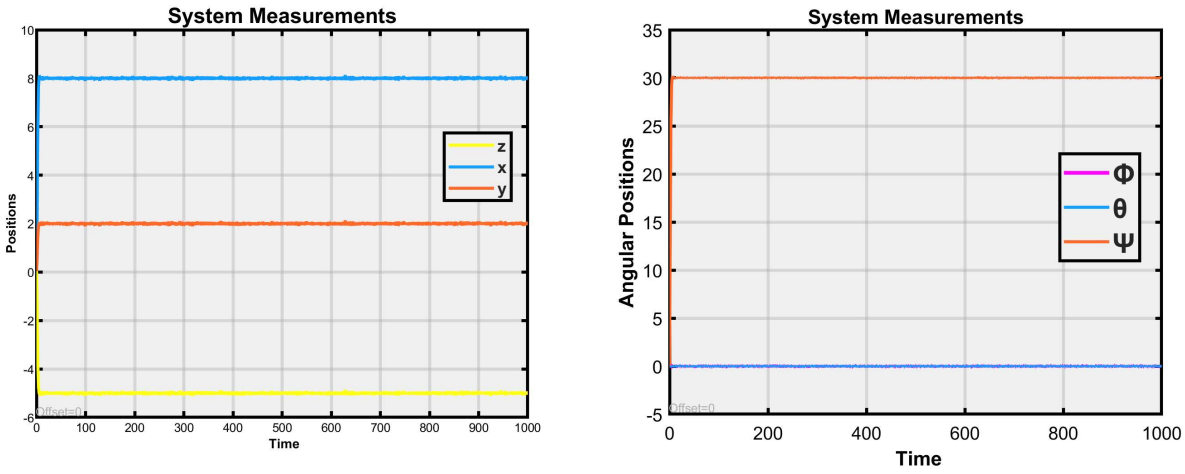

Figure 3.18: C&C side outputs without attack and coding-decoding effect.

Figure 3.18 illustrates the healthy functioning of the quadrotor system. In this scenario, the plant configuration remains consistent with that of the previous section. The reference signal, issued by the C&C, mirrors the conditions specified in Scenario A, with reference values set at $R = [30, 2, -5, 8]$, as discussed earlier. The UAV design aims to achieve set-point regulation, ensuring the system stabilizes around the specified reference points. Ideally, sensor measurements should accurately reflect these reference parameters. This figure demonstrates the position and angular position of the UAV for 1000 unit time of the simulation results.

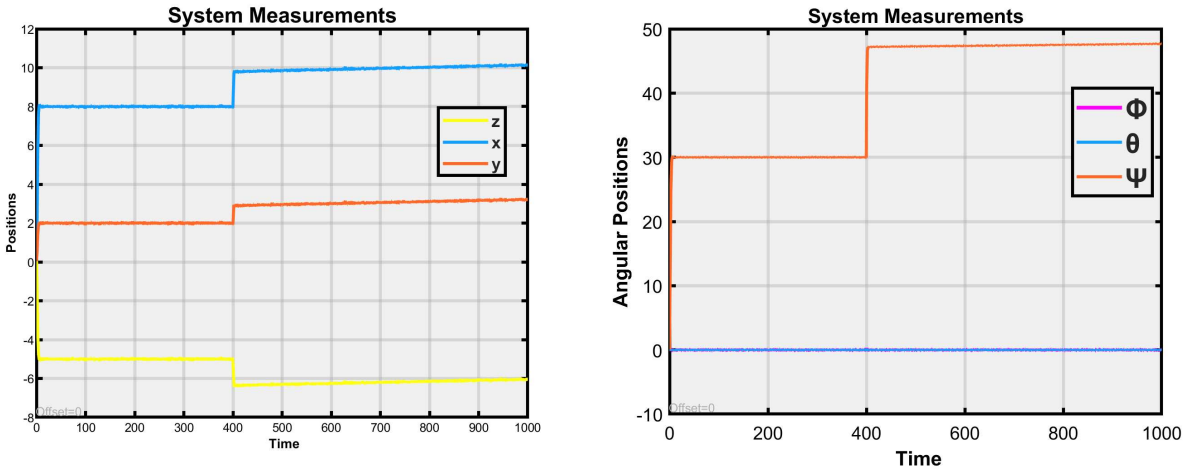

Figure 3.19: Plant side outputs with attack and without coding-decoding effect.

Figure 3.19 demonstrates the functionality of the system in the presence of a replay attack. The attacker initiates the attack at time 400 until the end of the mission. Consistent with the nature of a replay attack, there exists a period prior to time 400 during which the attacker records system data. This recording phase allows the attacker to replay previously captured data, thereby enhancing the stealth of the attack. In this compromised scenario, the C&C remains unaware of the altered system behaviour, displaying the operation as if unaffected. Consequently, the monitoring display within the C&C reflects the expected, unaffected behaviour of the system as follows:

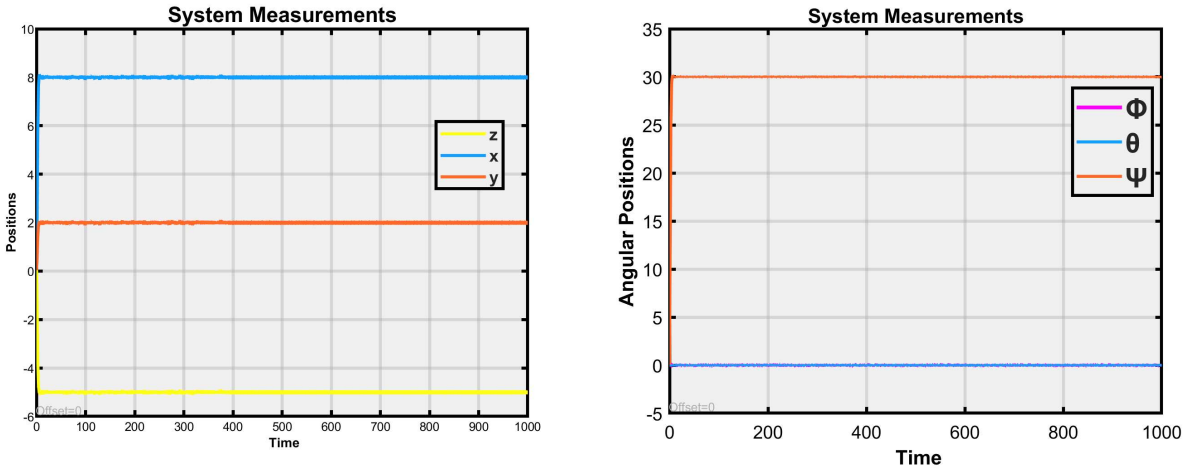

Figure 3.20: C&C side outputs with attack and without coding-decoding effect.

Figure 3.20 illustrates the monitoring system within the Command and Control (C&C) center during a replay attack. In this instance, the attacker attempts to inject previously recorded outputs into the

system periodically. However, since the UAV operates in a steady-state condition, the monitoring center fails to detect any attack within the process. As a result, the attack remains undetected by the C&C, allowing the injected data to go unnoticed and preserving the appearance of normal system functionality like Figure 3.18. The following parts represent the system operation from the plant side and the command and control center in the presence of the designed coding, respectively.

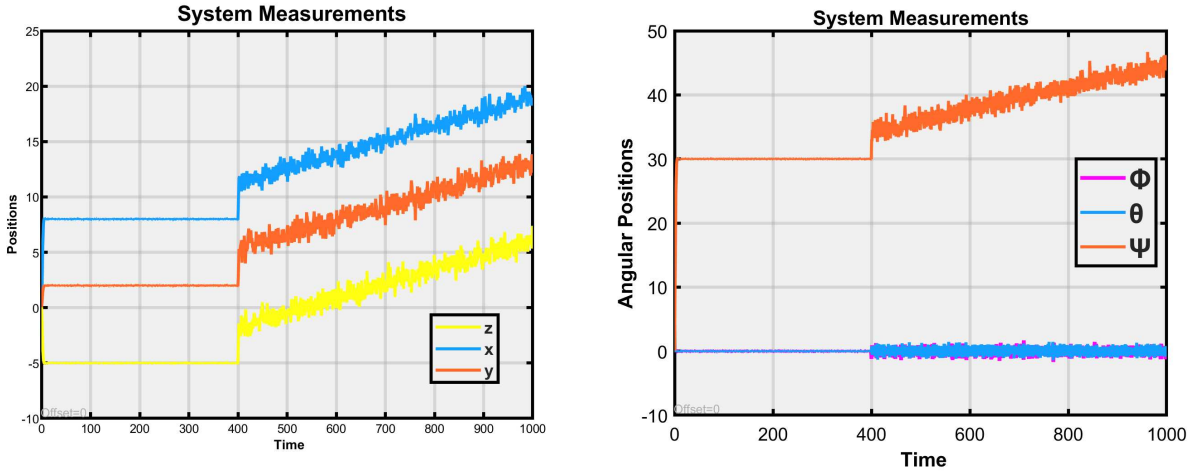

Figure 3.21: Plant side outputs with attack and coding-decoding effect.

Figure 3.21 represents the sensor outputs on the plant side in the presence of an attack, alongside the effects of the implemented coding design. The figure shows that the coding and decoding influence the sensor measurements. However, since this coding scheme has been applied from the start of the UAV's operation, it does not negatively impact the UAV's healthy functionality. The coding design introduces subtle changes in the sensor readings, which become apparent when the attack is active within the loop. Despite these deviations, the monitoring system within the C&C center is unable to observe or detect this variation. Consequently, the C&C displays the sensor outputs as follows:

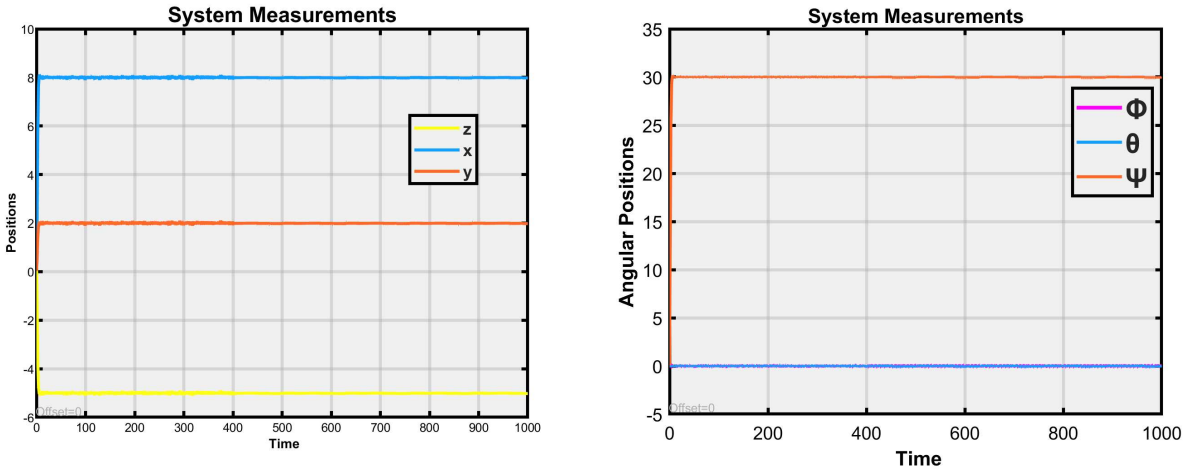

Figure 3.22: C&C side outputs with attack and coding-decoding effect.

Figure 3.22 illustrates the monitoring system within the C&C center to control and evaluate system functionality. As shown, this monitoring center continues to observe the periodic outputs injected by the attacker. However, due to the implemented coding-decoding design, certain portions of the recorded data are influenced by the coding effect, becoming apparent when the attacker is within the loop. Specifically, the periodic matrix changes over time, creating deviations that the C&C center can detect, making the attack potentially observable.

Nevertheless, this detectability relies on a critical assumption: if the attacker is capable of recording all possible coding-decoding variations contained within the dictionary during the recording phase and the recording window is sufficiently extended to capture all coding matrices—then the replay attack may remain undetectable. A chi-square detector is subsequently employed to assess this detectability, as demonstrated in the figure as follows:

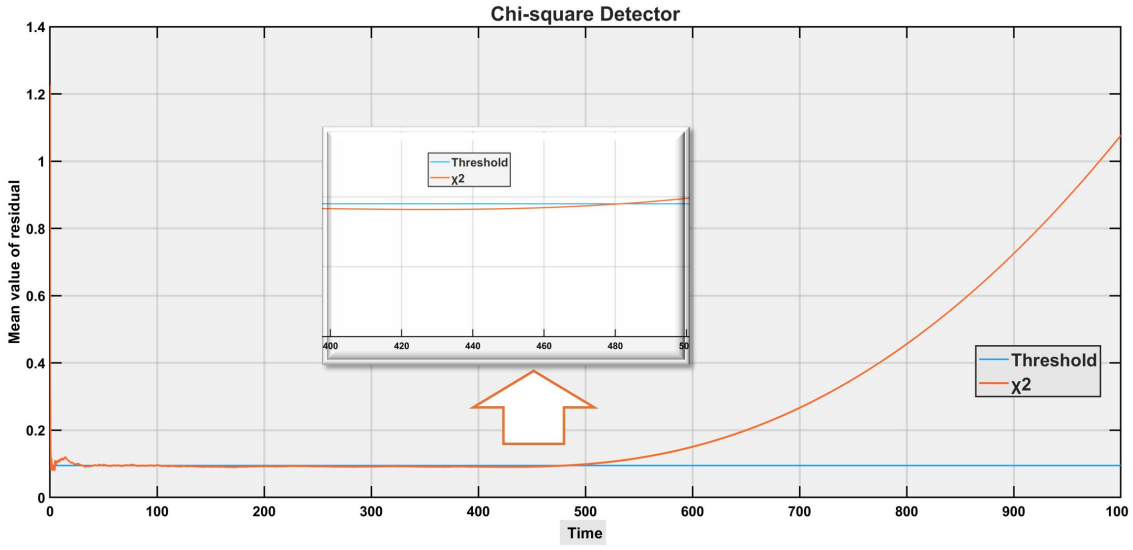

Figure 3.23: χ^2 test for replay attack scenario in the presence of coding-decoding.

Figure 3.23 presents the results of the chi-square test, evaluated against a predefined threshold selected based on the system's healthy and noisy conditions. As shown, subtle changes appear in response to replay attack detection, where the mean value of the residual is utilized for evaluation. This approach enhances the clarity of the variations in the attack pattern due to the nature of replay attacks, as the mean value of the residual generated by the chi-square detector crosses the predefined threshold at approximately time 480. Although this detection exhibits a delay relative to the attacker's injection time, which occurred at time 400, the decision-making center can ultimately identify the attack and raise an alert. This delayed yet successful detection highlights the effectiveness of the chi-square test in identifying replay attacks as it remained above the threshold.

The simulation results presented in this section demonstrate that the designed periodic coding-decoding technique effectively enhances the system's detection capabilities in the presence of both covert and replay attacks. This technique introduces vulnerabilities that allow the system to detect abnormal behaviours more effectively, improving overall system resilience against such attacks.

3.6 Conclusion

As a recap, this chapter began by defining the problem statement, which is the existence of stealthy cyber-attacks such as covert or replay within a CPS configuration, which requires a reliable technique to be able to detect these kinds of attacks. Following this, the model-based methodology was considered, and the proposed design was investigated step-by-step to achieve the setpoint regulation goal. Within this framework, the design of the LQI control was examined to ensure and guarantee the stability of the system during its mission. Additionally, the KF was designed to monitor the system throughout the mission for security purposes.

The detection and decision-making component was developed for the existing methodology. Subsequently, the milestone addressing the covert attack was explored through the design of the coding-decoding matrix, which had multiple objectives. Firstly, the decoding process is needed to ensure the system's guaranteed functionality during the mission while incorporating an external coding element. Secondly, the design required sufficient complexity to achieve security goals while remaining unidentifiable by attackers. This objective was addressed through the use of a periodic coding matrix. However, two main assumptions were necessary. First, the attacker could not access the predefined dictionary, and second, a secure channel existed for transferring the coding index from the C&C to the plant for the synchronization objective. Without considering these assumptions, stealthy attacks remain detectable until the attacker processes the dictionary and understands the possible matrices. However, as the entire matrix is not transmitted through the secure communication channel, the synchronization process becomes time-consuming. During this period, the attack remains detectable for a certain amount of time. The proposed solution is not only able to address detection concerns with respect to covert or replay attacks but also able to enhance the security of the entire system, which prevents attackers from posing an undetectable attack successfully. After providing mathematical steps, the evaluation of the effectiveness of the considerations is done through simulation results on the quadrotor with 6DoF. The results for each part were demonstrated using MATLAB simulations for two scenarios concerning the covert attack and one scenario for the replay attack detection.

As future directions, this research could be expanded by evaluating the enhanced detection tests not only on covert and replay attacks but also on other existing attack types documented in the literature. Another prospective direction would involve developing subsequent steps, such as identification and mitigation strategies, following the detection of these attacks, which are addressed effectively. Additionally, considering the assumptions in replay attack detection, further research could explore the impact of increasing the coding possibilities and expanding the coding-decoding design to improve detection effectiveness. From a security standpoint, testing the secure communication channel and the pre-defined dictionary with various encryption, decryption, and authentication techniques presents an opportunity to protect critical data against attackers, thereby ensuring the method's robustness in real-world applications.

Chapter 4

Data-Driven Covert and Replay Attack Detection and Identification

This chapter introduces neural network-based algorithms for detecting and identifying covert and replay attacks on the CPS framework when there is no access to the mathematical representation of the system. In this scenario, the only access the Command and Control (C&C) has is the received data from the output channel and the data sent from this center through the input communication channel, a collection of input-output data without a discernible pattern. The primary contribution of this chapter lies in the identification of covert and replay attacks, which were detected in the previous chapter, and effectively isolating these attacks using the proposed algorithm. Since this chapter adopts a data-driven approach, it also demonstrates the capability of the designed algorithm to detect each type of attack. However, the primary focus is not on the detection itself, as the detection of these attacks was already accomplished in the preceding chapter through a coding-decoding scheme. The dataset is extracted from the previous chapter to achieve its objectives. The primary objective of the designed algorithm is to perform feature extraction from the raw dataset before model development using a Neural Network (NN). This approach offers several advantages, including reducing the complexity required for the NN model to achieve the identification objective. Furthermore, it enhances model accuracy by focusing on the essential behaviours of the data, effectively extracting relevant features and relationships. By isolating critical features from the

raw dataset, the algorithm significantly reduces memory usage, enabling the model to use only the selected features as input for the NN, simplifying the model development process.

Section 4.1 carefully describes the problem statement and the basic idea to solve the existing problem in a case where only input-output data of the system is accessible without knowledge of the mathematical representation of the system. Also, explicitly exploring the best placement of these methodologies and suitable algorithms and procedures while having the CPS framework. Section 4.2 deeply explains the considered methodology in which subsection 4.2.1 provides the architecture which is efficient for the feature selection from the raw data set to address the detection of covert and replay attacks separately, and subsection 4.2.2 explores the procedures used to detect each type of attack individually and presents the NN model that effectively achieves the detection identification objectives. Finally, Section 4.3 provides a numerical example that visually illustrates the practicality of the proposed procedure and demonstrates the results of the designed algorithm within the CPS framework.

4.1 Problem Statement

From a data-driven point of view, the problem centers around handling large datasets without a mathematical system representation. In this context, the C&C unit only has access to raw data from sensor channels and the reference signal sent by the C&C itself. This situation demands highly accurate and robust algorithms capable of monitoring system behaviour and extracting relevant patterns directly from the received data to evaluate system behaviour. As mentioned earlier, a key challenge in CPS configurations is security, mainly when dealing with attackers who can gradually add to their knowledge or communicate with other attackers to strengthen their capabilities. The primary concern is that such attackers may initially have access to resources, enabling them to disrupt and disclose sensitive data through replay attacks at specific times during a mission. As the attacker gains the system knowledge, they are able to carry out more sophisticated and harmful attacks by implementing covert attacks, posing a significant threat to system integrity and security. The other remaining challenge lies in developing an algorithm that is not only capable of detecting each type of attack based on the distinct effects they impose on the data but also able to identify

and isolate these two different types of attacks. Achieving this distinction is crucial, as it provides a foundation for more effective handling and mitigation in subsequent steps.

Figure 4.1 represents the problem visually. At the same time, the intelligent attackers exist in the CPS configuration, and the C&C has only access to the data transmitted through the communication channel, which is vulnerable to attacks.

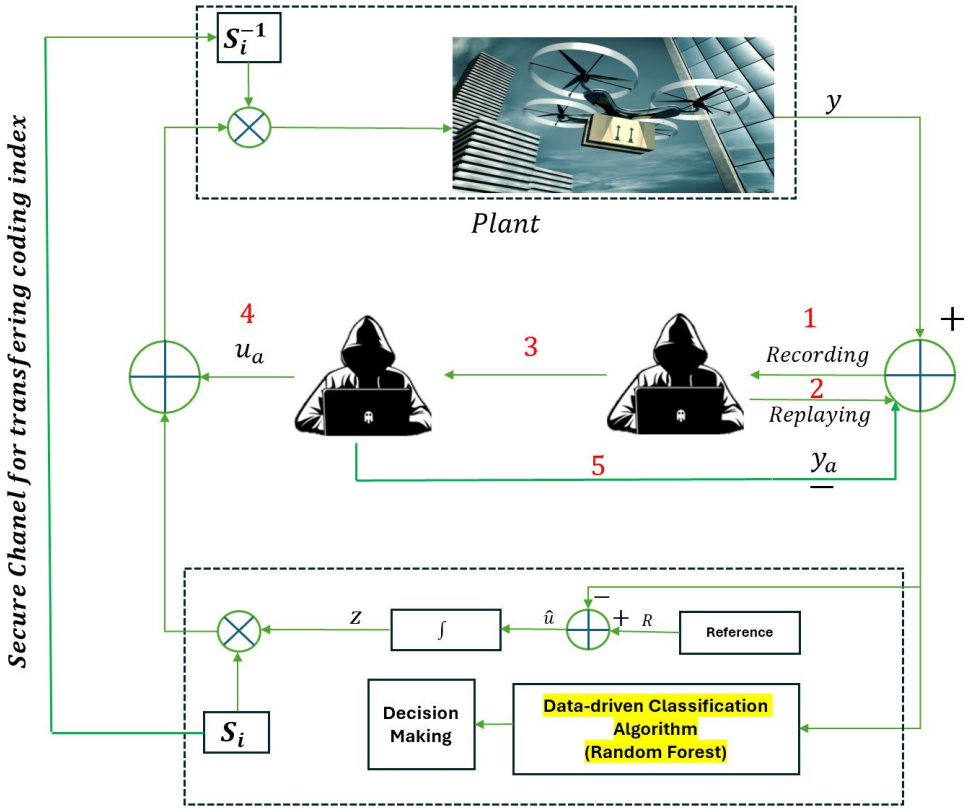

Figure 4.1: Overview of the problem II.

As illustrated in Figure 4.1, the existing algorithm is positioned within the C&C unit, replacing the partial controller, estimator, and observer used in the previous chapter for comparison.

4.1.1 Problem Definition & Assumptions:

The core issue addressed in this research is the vulnerability of a linearized quadrotor, modelled as a CPS configuration, to covert and replay attacks. These attacks, each affecting the data in distinct patterns, necessitate proper isolation and identification due to their undetectable nature. A refined algorithm is proposed to address these sophisticated threats, aiming to accurately identify and distinguish between these attacks. The solution used the Random Forest(RF) algorithm to analyze the hierarchical dataset, enhances feature selection and reduces the complexity of the NN algorithm used for model design and system performance evaluation. The designed algorithm can detect scenarios in which the system deals with any of these attacks separately. Each existing decision tree can assist in ranking and determining the importance of input data for a neural network, as this algorithm can identify the relative importance of various features in predicting a target. By utilizing a metric such as the Gini index, the decision tree identifies the features that significantly impact data classification. Consequently, the decision tree algorithm first detects which data points, for instance, have the greatest influence on classification and assigns them a weight of importance, which is then passed to the neural network to expedite its classification process. This approach could significantly contribute to the detection of attacks. This information (the importance of inputs and their corresponding weights) helps the neural network to focus on more relevant features, thereby reducing model complexity.

Assumptions: The first assumption is that the data extraction process is grounded in the findings and methodologies developed in the previous chapter, ensuring consistency and reliability in how data is handled. This extracted data has the influence of coding and decoding matrices applied within the system, capturing the essential characteristics and behaviours necessary for analysis. The second assumption stands as the algorithm designed for this study uses the system's actual output, which is transmitted to the C&C center, as its primary input for the feature selection process. This ensures that the algorithm operates on real-time data, allowing for accurate monitoring, detection, and identification of potential attacks within the system.

In the upcoming section, the designed algorithm and its mathematical formulation for feature selection will be examined in detail, step by step. Feature selection is a key process, playing a crucial

role in simplifying the artificial model design steps and enhancing model accuracy in subsequent stages.

4.2 Proposed Methodology

The proposed algorithm in this study is designed to address the detection and isolation of covert and replay attacks in CPS through an integrated approach combining feature selection and neural network techniques. By utilizing a robust data processing pipeline, key features are extracted and selected using the Random Forest algorithm, ensuring that only the most relevant parameters from the dataset are used. This enhances the accuracy of the system in detecting subtle variations between covert and replay attacks, which are often challenging to identify due to their similar behaviour patterns. The model integrated a multi-layer Artificial Neural Network (ANN), specifically structured with 64 neurons in the first hidden layer and 32 neurons in the second, to analyze the selected features. The ANN applies the Rectified Linear Unit (ReLU) activation function to capture relationships within the data, enabling precise detection of complex attack patterns.

The core strength of this algorithm lies in its ability to handle datasets containing both covert and replay attacks simultaneously while distinguishing them from normal system operations. By calculating the Gini impurity during the decision tree construction process, the Random Forest ensures optimal feature selection, improving classification performance. Additionally, the model's training process, supported by the Adam optimization algorithm, ensures rapid convergence and accurate predictions, even when the dataset includes overlapping attack scenarios. This makes the algorithm highly effective in real-time environments where the system is vulnerable to both types of attacks. The ability to not only detect but also isolate covert and replay attacks from healthy operational states highlights the power and adaptability of the proposed solution, offering a reliable defence mechanism for CPS environments. This section will examine the steps involved in designing the algorithm, providing a detailed, step-by-step analysis of the process.

4.2.1 Data Processing and Feature Selection Using Random Forest Algorithm:

Data processing is a fundamental step before proceeding to feature selection and extraction from raw data. Understanding and evaluating the nature and behaviour of the data ensures that learning models perform optimally, reducing noise and enhancing predictive accuracy. Data prepossessing encompasses various tasks to clean and transform raw data into a format suitable for models. This section outlines the key steps in data processing and feature selection from the raw data set.

Data Loading:

The initial step is data loading, where raw data is retrieved and prepared for further analysis. In the context of CPS, the data may include normal operational behaviour and attack scenarios. Loading this data correctly ensures that all relevant features are available for subsequent steps, and issues like missing values, formatting inconsistencies, and data integrity must be checked and resolved at this stage. The data set is represented as follows:

$$D \in R^{m \times n} \tag{82}$$

D is the complete dataset, and n denotes the number of input parameters.

Investigation and Removal of Unnecessary Features By Random Forest:

Raw parameters are not informative enough to contribute to the learning process, which requires investigation to enhance model performance and reduce overfitting concerns. These redundant parameters correspond to the input data in the existing problem. This new data set is represented as follows:

$$D^{'} = \{f_1, f_2, ..., f_k\} \tag{83}$$

Equation (83) demonstrates the parameters are directly involved in the process and possess significant characteristics for ensuring that the dataset is sufficiently informative for the investigation.

Identification of Important Features:

Identifying important features improves the model's efficiency and accuracy. This step involves evaluating each parameter's contribution to the target labels. It is especially important in attack detection and identification within CPS, where the influence of specific relations of parameters, which can also call features, can differ significantly.

A standard method for feature importance is using decision tree-based algorithms, such as the Random Forest (RF). This algorithm generates multiple decision trees and calculates each feature's importance by assessing how often it is used in splits that reduce impurity. RF falls under the category of bagging methods. This approach enhances predictive performance by constructing multiple decision trees and aggregating their predictions. Figure 4.2.1 represents the single decision tree as a type of supervised learning algorithm, which is mostly usable for classification and regression tasks.

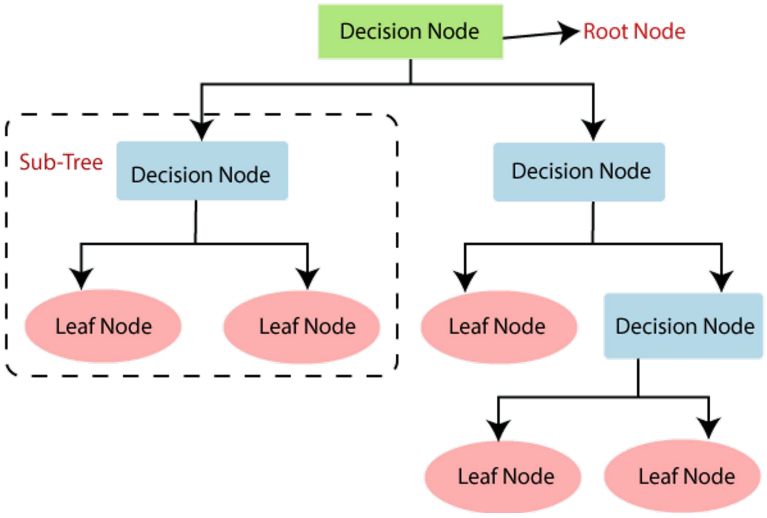

Figure 4.2: Decision tree algorithm.

Decision Trees(DT) Construction:

During the construction of each decision tree, the bootstrap aggregation (bagging) technique is utilized for sampling data. A new sample is drawn from the data with replacement. Each sample consists of N selections, some of which may be selected multiple times. Here, assumed the redefined dataset $D^{'}$ which explained in Equation (83) contains N samples as follows:

$$D'' = \{(x_1, y_1, z_1, \phi_1), (x_2, y_2, z_2, \phi_2), \ldots, (x_n, y_n, z_n, \phi_n)\} \tag{84}$$

Where represents the samples drawn with replacement from D''. For each bootstrap sample, a decision tree is constructed. The decision tree uses the Gini Impurity(GI) to classify the data. Here, the dataset at node t has a probability p_c for class c. The GI is mathematically defined as follows:

$$G(t) = 1 - \sum_c {p_c}^2 \tag{85}$$

During each node split, the feature ΔG that results in the maximum reduction of GI is selected for the split as follows:

$$\Delta G(t) = G(t) - \frac{n_L}{n} G(t_L) - \frac{n_R}{n} G(t_R) \tag{86}$$

Where n_L and n_R are the number of samples in the left and right nodes, respectively. The tree continues to grow until all samples in the nodes are identical (resulting in zero Gini impurity) or the number of samples reaches a level at which further splitting of the time series data is no longer beneficial for the trained model.

The Gini Index measures impurity in decision trees, showing how well parameters can separate the data into different classes. Gini impurity is calculated for each parameter at the current node and its child nodes. The other representation of Equation (86) is computed as follows:

$$\Delta G = G(\text{parent node}) - G(\text{left node}) - G(\text{right node}) \tag{87}$$

The parameter that results in the highest reduction in impurity is selected for the split. This process is repeated for the child nodes, helping the tree identify the most important features. Features that lead to a greater reduction in Gini impurity gain more importance, are placed higher in the tree, and play a more significant role in classification, while less important features might not be used or appear lower in the tree. This feature selection method helps improve model performance and accuracy.

One of the main drawbacks of using a decision tree in complex scenarios is its performance degradation with large datasets. The algorithm often struggles to maintain accuracy when faced with large volumes of data or random changes in multiple input variables. Additionally, decision trees may prove ineffective when dealing with complex classification tasks, as they can easily over-fit or under-fit the data. The RF algorithm generates multiple decision trees and calculates each feature's importance by assessing how often it is used in splits that reduce impurity, thus improving overall model accuracy. Figure 4.3 represents the RF as follows:

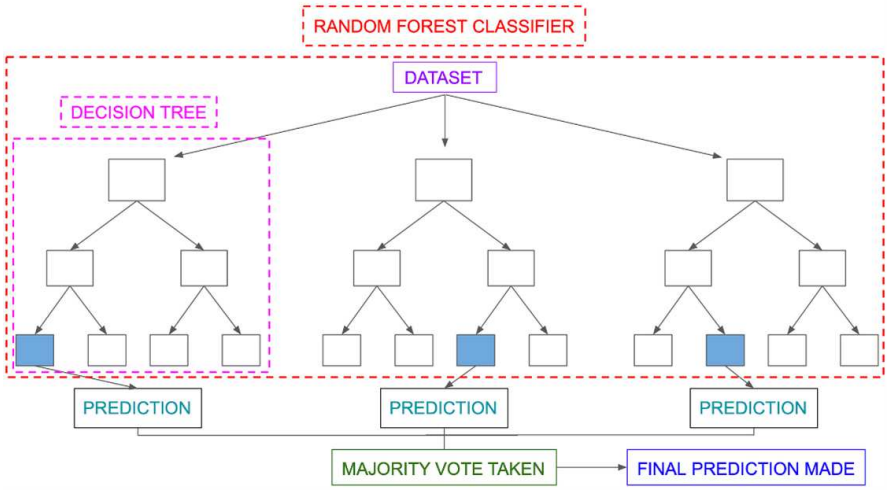

Figure 4.3: RF algorithm with fixed random states.

As shown in Figure 4.3 after constructing $t^{'}$ decision trees, the final prediction is made using majority voting. Let $\hat{y}$ represent the final output of the model and $\hat{y}_i$ be the output of the i-th tree. The final prediction $\hat{y}$ is computed as follows:

$$\hat{y} = mode\{\hat{y}_1, \hat{y}_2, \ldots, \hat{y}_{t'}\} \quad (88)$$

Where mode is the function that returns the most frequent (or majority) prediction from the set of trees. This voting mechanism ensures that the final prediction is robust, benefiting from the collective predictions of all trees. In other words, the mode of the outputs from the different trees is considered the final prediction, leading to the selection of the best features.

In this phase, each dataset is divided into two parts: a training set and a validation set. The training

set is used to fit the model, while the validation set is reserved for evaluating the model's performance during the training process. A typical split involves allocating 80% of the data for training and 20% for validation. This approach allows the model to learn from the majority of the data, ensuring it captures the essential patterns. At the same time, the remaining portion is utilized to assess the model's generalization ability. By setting aside a validation set, the model's performance on unseen data can be more accurately evaluated, helping to prevent overfitting and ensuring that the model is robust when applied to new, unseen datasets.

4.2.2 Structure of the Artificial Neural Network(ANN) Model:

A typical ANN model usually consists of several layers, each containing a specific number of neurons. Regarding the structure of the neural network, the first layer of the network defines the input to the model. In this phase, the processed features are fed into the model as input. Each feature corresponds to one input neuron. The second (hidden layer) is responsible for learning more complex behaviour from the selected features. The first and second layers are 64 and 32 neurons, respectively. The linear transformation for the first input layer is shown as follows:

$$z^{(1)} = W^{(1)}x + W^{(1)}y + W^{(1)}z + W^{(1)}\psi + b^{(1)} \tag{89}$$

Where $z^{(1)}$ is the pre-activation function of the first hidden layer,$W^{(1)}$ represents the weights associated with each input feature at the first hidden layer, and $b^{(1)}$ is the bias term introduced to this layer to ensure that the model can fit the data well. This non-linear transformation prepares the input features for the non-linear active action function that will be applied next in the hidden layer to introduce non-linearity into the model, enabling it to capture complex patterns in the data.

The Rectified Linear Unit (ReLU) activation function introduces non-linearity into the model. It is chosen due to its high efficiency in neural networks, especially in classification tasks. The formulation of the ReLU is shown as follows:

$$A^{(1)} = \text{ReLU}(Z^{(1)}) = \max(0, Z^{(1)}) \tag{90}$$

Where $a^{(1)}$ is the output of the first hidden layer after applying the ReLU activation function. As

mentioned earlier, the second hidden layer has 32 neurons, which the $A^{(1)}$ serves as input of the second hidden layer. The linear transformation function of this layer is shown as follows:

$$Z^{(2)} = W^{(2)}A^{(1)} + b^{(2)} \tag{91}$$

Applying the ReLU activation function in the second hidden layer is represented as follows:

$$A^{(2)} = \text{ReLU}(Z^{(2)}) = \max(0, Z^{(2)}) \tag{92}$$

Where $W^{(2)}$ represents the weights for the second hidden layer, $b^{(2)}$ is the bias term for the second hidden layer,$Z^{(2)}$ is the pre-activation output of the second hidden layer, and $A^{(2)}$ is the final output after applying the ReLU activation function.

The ReLU function is applied to introduce non-linearity into the model, allowing it to learn complex patterns from the input data. The output $A^{(2)}$ is passed to the subsequent layers for further processing. The last Layer, the Output Layer, is responsible for the final decision-making. In this case, a dense layer with two neurons is used, as the problem under consideration is binary classification. The Softmax activation function is applied in this layer to compute the probabilities of the classes and perform the final binary classification. The following figure demonstrates the suggested model visually as follows:

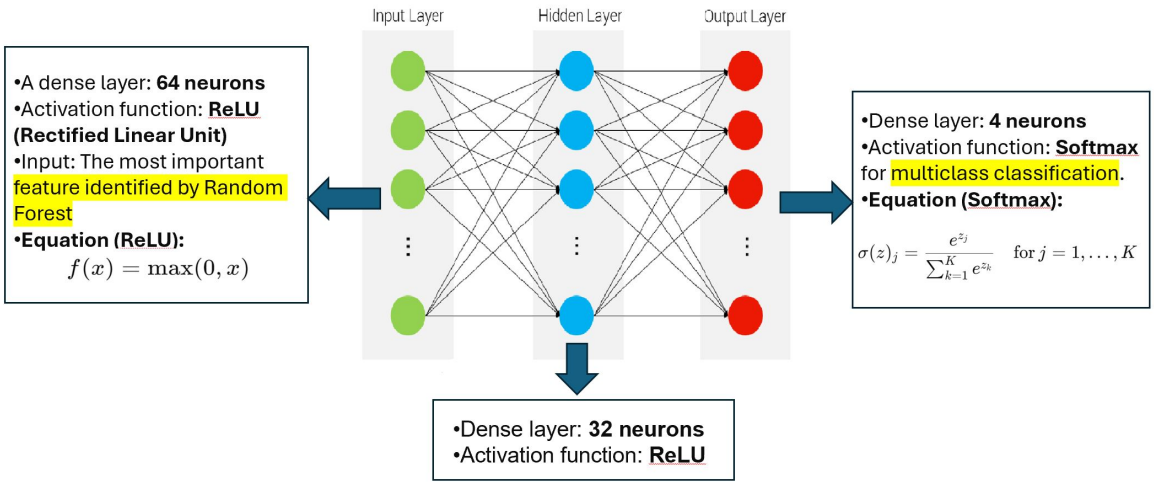

Figure 4.2.2 demonstrates the considered model configuration for the covert attack detection goal as a part of this research goal.

The output layer is responsible for the final decision-making. In this case, a dense layer with two neurons is used, as the problem under consideration is a binary classification (success or failure of an attack). The Softmax activation function is applied in this layer to compute the probabilities of the classes and perform the final classification. The linear transformation for the output layer is defined as:

$$Z^{(3)} = W^{(3)} A^{(2)} + b^{(3)} \tag{93}$$

In Equation (93), $W^{(3)}$ be the weight matrix for the output layer and $b^{(3)}$ be the bias term for the output layer. The Softmax activation function is applied to obtain class probabilities as follows:

$$P(class_i) = \frac{e^{Z_i^{(3)}}}{\sum_{j=1} e^{Z_j^{(3)}}} \tag{94}$$

Where $P(class_i)$ is the probability of class i from the softmax function , and $Z_i^{(3)}$ is the logic for class i. This transformation ensures that the output is converted into class probabilities, where each class is assigned a probability between zero and one, and the sum of all class probabilities equals one. The class with the highest probability is selected as the final prediction.

Model Training Equations :

For training the model, balanced and normalized selected features are utilized. The training process involves several key steps: loss and optimizer function selection. The sparse categorical cross-entropy loss function is suitable for multi-class or binary classification problems. It allows the model to improve its parameters by calculating the distance between the model's output and the actual label values. The sparse categorical cross-entropy function is given as follows:

$$Loss = -\sum_{i=1}^{2} y_i \log(P(class_i)) \tag{95}$$

Equation (95) shows the sparse categorical cross-entropy loss, y_i is the actual label of the sample i.

Regarding the selection of an optimizer, the Adam algorithm is used for optimization. This is one of the most potent and popular optimizations due to its unique features, such as high convergence speed and dynamic adjustment of learning rates, which are suitable for real-time systems. The update rule for Adam is shown as follows:

$$\theta_t = \theta_{t-1} - \frac{\eta}{\epsilon + \sqrt{\hat{v}_t}} \hat{m}_t \tag{96}$$

Equation (96) represents the used optimization technique where θ_t is the value of the weights of the network at iteration t, θ_{t-1} is the value of the weights at the previous iteration $(t-1)$, $\hat{m}_t$ is the bias-corrected first-moment estimate (mean of the gradients), $\hat{v}_t$ is the bias-corrected second-moment estimate (mean of the squared gradients), η is the learning rate, which determines the step size for each update, and ϵ is a small constant to prevent division by zero.The parameters $\hat{m}_t$ and $\hat{v}_t$ are calculated as follows:

$$\hat{m}_t = \frac{m_t}{1 - \beta_1^t}, \quad \hat{v}_t = \frac{v_t}{1 - \beta_2^t} \tag{97}$$

Here, m_t is the exponentially weighted moving average of the gradients, v_t is the exponentially weighted moving average of the squared gradients, and β_1 and β_2 are the decay rates for the moving averages. These weights are optimized throughout the training process to achieve the best accuracy for the model.

The primary evaluation metric is **accuracy**, which indicates the percentage of correctly classified samples, which is mathematically shown as follows:

$$\text{Accuracy} = \frac{\text{Number of correct predictions}}{\text{Total number of predictions}} \tag{98}$$

Additionally, the **loss** is calculated for both the training and validation sets as follows:

$$\text{Loss} = -\frac{1}{N} \sum_{i=1}^{N} y_i \log(\hat{y}_i) \tag{99}$$

The loss function helps determine how closely the model's predictions match the actual values, guiding the model to make more precise predictions.

Model Training Process

The model is trained on the training dataset through multiple iterations (epochs). In each epoch, the neural network minimizes error and improves the accuracy of its predictions. After each training epoch, the model is evaluated for its accuracy on the validation data to assess its performance on unseen data. The number of epochs is adjusted based on the model's convergence and the project's requirements. The training process steps are listed as follows:

(1) The training data is fed into the model.

(2) The model continuously adjusts weights and biases through the Adam optimizer and the loss function to improve its accuracy.

(3) At the end of each epoch, the model's accuracy and loss are calculated and recorded on the validation data as follows:

$$\text{Accuracy}_{\text{epoch}} = \frac{\text{Correct Predictions}}{\text{Total Validation Samples}} \tag{100}$$

(4) If the accuracy of the validation data exceeds previous results, the model is saved.

Regarding the final evaluation, the trained model is tested on the validation data, and the results are recorded. The best model is selected and saved based on the highest accuracy achieved during validation. The saved model can then predict and classify new samples in future applications.

4.2.3 Recap of the Methodology Steps:

The proposed methodology has been developed through a detailed, stepwise mathematical framework. The algorithm addresses the core processes and critical computational steps by systematically addressing each aspect of the problem. In particular, this algorithm summarizes the essential procedures, integrating previously elaborated theoretical concepts. Its design aims to facilitate the efficient execution of the proposed approach, ensuring clarity and precision in its operations. The algorithm serves as a guide to the methodology, offering a streamlined view of the implementation steps, which will be demonstrated in the subsequent numerical example.

Algorithm 2 Feature Selection and NN Model Structure

Feature Selection
Step 1: Load Data
for each file in the dataset **do**
    if file is valid **then**
        Load data into memory
    else
        Print ”Invalid file” and skip
    end if
end for
Step 2: Identify Important Features
for each parameter in the valid dataset Calculate importance using Random Forest **do**
    if feature importance $\geq$ threshold **then**
        Mark feature as unimportant
    else
        Keep feature
    end if
end for
Step 3: Split Data into Training and Validation Sets
Split data into 80% training and 20% validation sets
Step 4: Balance Data using SMOTE
while dataset is imbalanced: **do**
    Apply SMOTE to generate synthetic samples & check the balance of classes
end while
Model Structure
Step 1: Define Model
    Define the input layer with one neuron per feature
    Define hidden layer 1 with 64 neurons and ReLU activation
    Define hidden layer 2 with 32 neurons and ReLU activation
    Define the output layer with 2 or 4 neurons and Softmax activation function depending on the classification goal
Step 2: Train Model
for each epoch in training: **do**
    Pass training data through the model Calculate loss using sparse-categorical-cross-entropy Update weights using Adam optimizer
    if validation accuracy improves: **then**
        Save the model
    else
        Skip
    end if
end for
Step 3: Evaluate and Save Model
Test the trained model on validation data
if validation accuracy is highest: **then**
    Save the final model
end if

4.3 Numerical Example:

In this section, the simulation results are presented to evaluate the performance and robustness of the proposed algorithm for detecting cyber-attacks in UAVs. The goal is to demonstrate the model's effectiveness in identifying covert and replay attacks under various conditions. The data sets used in these simulations are derived from the experiments described in the preceding chapter and have been carefully designed to encompass a wide range of operational scenarios, attack types, and parameter variations.

The simulations are structured into three primary scenarios. The first scenario focuses on covert attacks, where five distinct data sets are analyzed, each containing attacks occurring at different time intervals. The second scenario assesses the algorithm's capability to detect replay attacks using five additional data sets with characteristics unique to replay attack behaviours. Finally, the third scenario includes an evaluation involving 30 data sets, each characterized by diverse parameters such as varying attack injection times, reference signals from the C&C, and recording periods of the replay attack. These parameters simulate different UAV missions and operational profiles, testing the model's generalizability and adaptability to changing conditions.

The data sets utilized in these simulations consist of time-series data that captures the UAV's operational sensor outputs, including time, position, and angular positions (attitude), as recorded by the sensor channels in the C&C. As discussed in earlier chapters, the UAV is linearized and modelled with six degrees of freedom (6-DOF), and the extracted data effectively represents its dynamic behaviour. The simulation results aim to validate the algorithm's ability to detect attacks accurately and isolate their occurrence within the data sets, thereby demonstrating its applicability to real-world CPS security contexts.

The outcomes of the simulations provide evidence of the model's capacity to accurately identify the onset and nature of attacks while also showcasing its ability to maintain performance across varying conditions. This analysis underscores the algorithm's potential for enhancing UAV security and highlights its broader implications for ensuring the safety and resilience of CPS environments.

4.3.1 Covert Attach Detection Results

The first phase of the model's development focuses on detecting covert attacks. The goal of this evaluation is to demonstrate the model's ability to accurately identify covert attacks which compromise the system's security. In this context, the model undergoes systematic pre-processing, feature selection, and data balancing to ensure optimal detection performance. The flowchart provided effectively outlines each step in this process, depicting how data is processed and transformed from initial loading to final model training and validation. This structured approach ensures the model can detect covert attacks across various scenarios, enhancing its applicability and reliability in real-world applications. Figure 4.4 demonstrates these steps visually as follows:

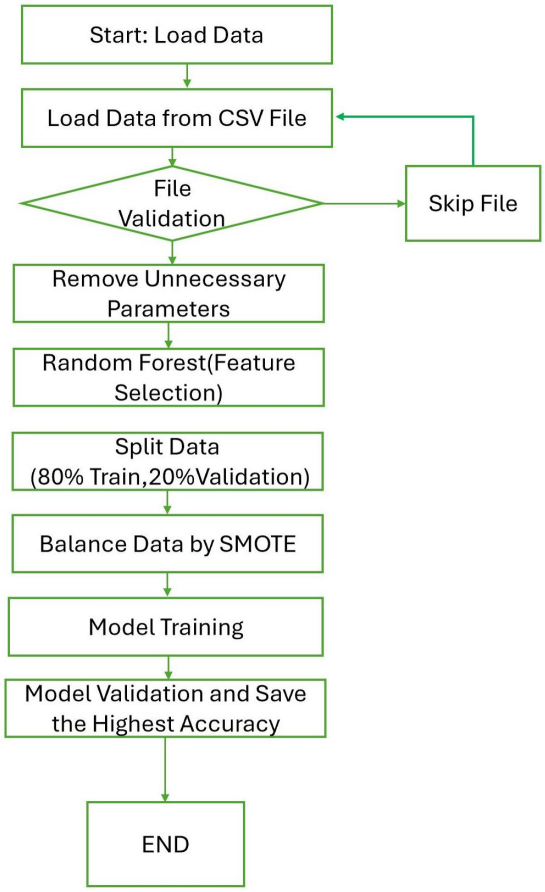

Figure 4.4: step-wise flowchart for covert attack detection.

Initially, data is loaded from CSV files, followed by identifying and removing unnecessary features.

Then, essential features are determined using Random Forest techniques. The data is split into training and validation sets, and SMOTE is employed to balance the classes in the training data sets. Finally, the prepared data is used to train an NN model, and the model is optimized using the Adam optimizer. During the training process, the model's accuracy is evaluated on validation data, and the best-performing model is saved.

In this section, the performance of the covert attack detection model during training is analyzed. Two key metrics, accuracy and loss, were used to evaluate the model's learning progress and generalization capability over time. These metrics were measured for the training and validation datasets across several epochs. The accuracy and loss validation are shown as follows:

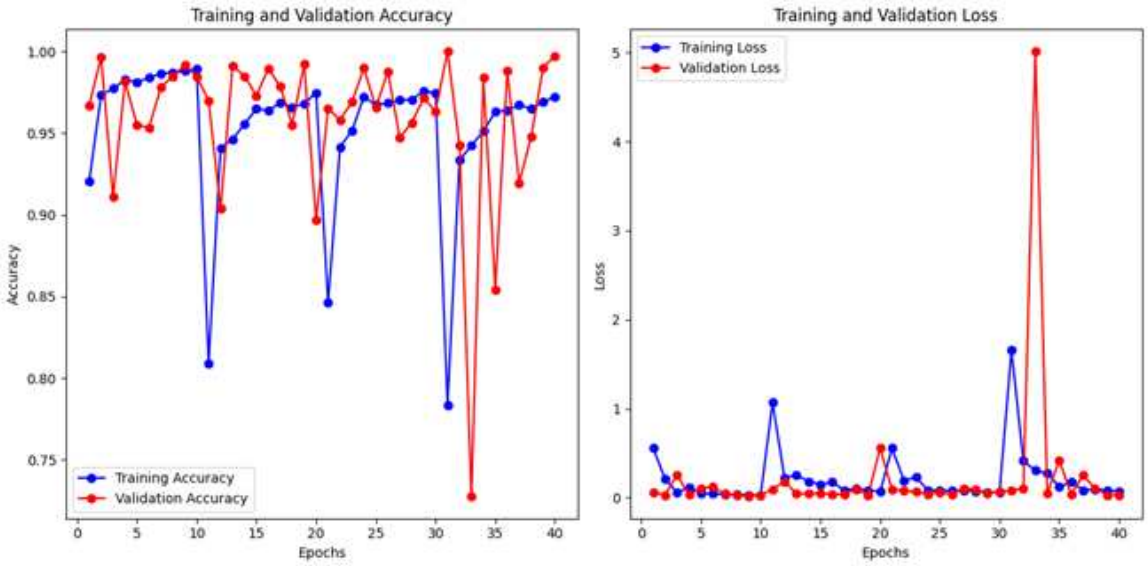

Figure 4.5: The accuracy and loss validation of the model in covert attack scenarios.

In both sub-figures, the blue line represents the performance of training data over 40 epochs, which accounts for 80% of each dataset and is selected randomly. In contrast, the red line represents the performance the validation data, comprising the remaining 20% of the dataset, with no overlap with the training data. In the left sub-figures, the horizontal axis represents the number of training epochs, and the vertical axis shows the accuracy of the model. As training progresses, the accuracy for both training and validation datasets increases steadily, with the model approaching a high accuracy level close to 100%. The blue curve represents the model's accuracy on the training data, while the red curve reflects the validation accuracy.

During the early epochs, the model demonstrates rapid learning, with both training and validation accuracies increasing sharply. However, fluctuations in validation accuracy are observed, particularly around epochs 15, 30, and 35. These fluctuations suggest challenges in the model's ability to generalize to the validation data, potentially due to the existing noise in the dataset despite these fluctuations. In the right sub-figure, known as the loss, the horizontal axis again represents the number of epochs, while the vertical axis shows the loss values. The loss metric measures how well the model's predictions match the actual outcomes, with lower values indicating better performance. At the start of the training process, the training and validation losses decrease rapidly, indicating that the model is learning effectively. However, similar to the accuracy plot, fluctuations in validation loss are observed, particularly around epochs 30 and 35, where the loss increases noticeably. These increases in loss coincide with drops in validation accuracy, suggesting that the model struggles to generalize to new data in some cases, especially during covert attack scenarios. Despite these fluctuations, the model is able to recover and reduce the loss over time, showing its capacity to adapt and improve throughout the training process.

The confusion matrix is another critical tool for evaluating the performance of classification models by assessing how well the model predicts different classes based on accurate labels. In this case, the matrix displays the classification results for classes Class 0 and Class 1. The following figure represents the confusion matrix of the existence model and how successful the model is in this matter.

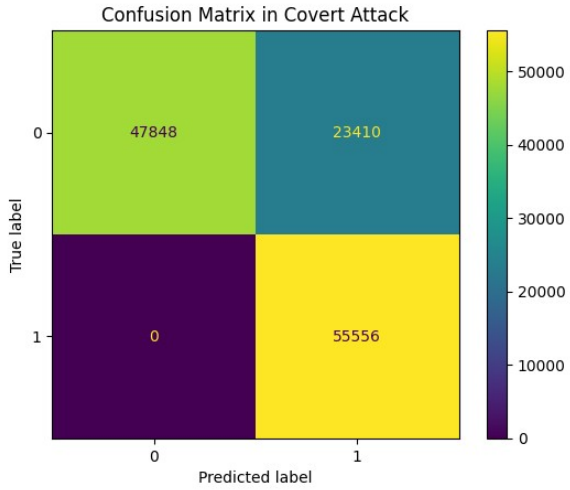

Figure 4.6: Confusion matrix of the model regarding covert attack detection.

Figure 4.6 provides insight into the model's performance in classifying covert attacks. 47,848 samples were correctly classified as Class 0, representing the True Negatives (TN). However, 23,410 samples were misclassified as Class 1 despite their true label being 0, indicating many False Positives (FP). Interestingly, the model did not incorrectly classify any samples from Class 1 as Class 0, as reflected by the absence of False Negatives (FN). On the other hand, 55,556 samples were correctly identified as Class 1 as True Positives (TP).

The Receiver Operating Characteristic (ROC) curve is a tool for evaluating the performance of classification models across various threshold levels. It plots the True Positive Rate (TPR) against the False Positive Rate (FPR) to visually represent the trade-off between sensitivity and specificity. The TPR, also known as sensitivity, measures the model's ability to correctly detect positive samples (Class 1). At the same time, the False Positive Rate represents the model's error rate when negative samples (Class 0) are incorrectly classified as positive. The ROC curve assesses how well the model distinguishes between classes, and the Area Under the Curve (AUC) quantifies the overall performance, with higher AUC values indicating better classification capability. The following figure illustrates the ROC curve, evaluating the model's performance in covert attack detection.

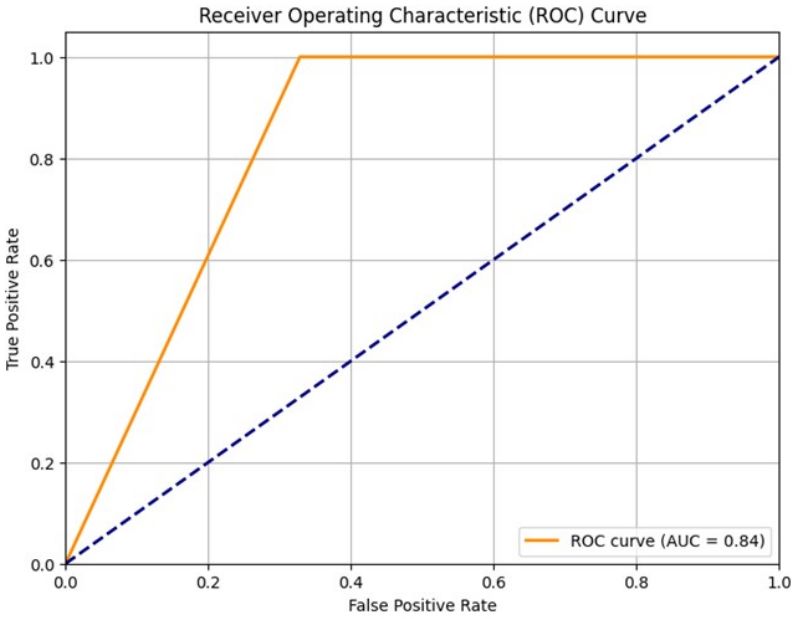

Figure 4.7: Model performance analysis concerning covert attack detection.

Figure 4.7, the blue diagonal line represents random performance, with an AUC value of 0.5, indicating no ability to distinguish between the classes. In contrast, the orange curve depicts the actual performance of the model, which achieves an AUC of 0.84. This value reflects the model's ability to correctly classify positive and negative samples and suggests that the model performs well in detecting covert attacks, but there remains room for improvement.

The "True Labels vs. Predicted Labels" chart provides a comparison between the actual labels (True Labels) and the labels predicted by the model (Predicted Labels) in a covert attack detection scenario. This chart highlights the model's effectiveness in correctly predicting class labels by showing how well the predicted labels align with the true labels. The designed model is powerful in datasets with covert attacks. Additionally, the integration of machine learning techniques such as Random Forest enhances accuracy by combining multiple decision trees to avoid overfitting, while SMOTE helps balance class distributions, further improving detection. For researchers working in real-time missions, such a comparison between true and predicted labels is vital for evaluating the model's reliability and accuracy in real-world applications. The following figure demonstrates this comparison, providing an overview of how the model performed in the covert attack detection task.

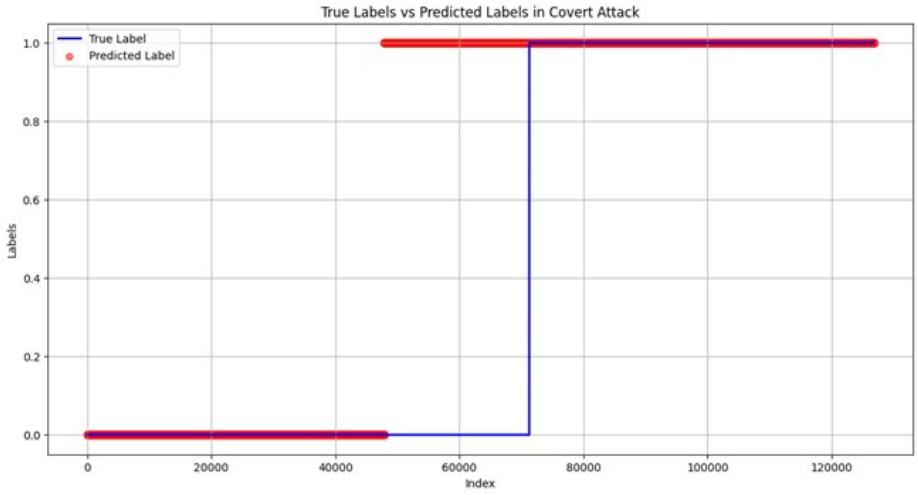

Figure 4.8: True label Vs. Predicted label from suggested model.

In Figure 4.8, the red points represent the labels predicted by the model. It is observed that the predicted labels align well with the true labels, indicating that the model has performed effectively in detecting covert attacks. True Label indicates the actual labels for each sample. At the beginning of the dataset, most samples belong to Class 0 (no attack or normal behaviour). However, a change occurs around indices 55,000 to 60,000, where most samples belong to Class 1 (covert attack).

4.3.2 Replay Attack Detection

The replay attack is considered a subset of the covert attack within the detection strategy; it presents specific patterns due to a lack of system knowledge that is critical for the designed algorithm to learn and recognize the pattern. The repetitive nature of the replay attack creates distinctive data patterns, enabling the algorithm to not only detect the replay attack itself but also improve its ability to identify between replay and covert attacks. By learning these patterns, the algorithm can better isolate and address the nuances between these two types of attacks, which share similarities but exhibit minor differences. This process is essential for enhancing the robustness of the detection system. To illustrate the nature of the data that the C&C receives during an attack, the presents sample datasets from four training scenarios. These datasets offer insight into the type of data the algorithm processes and how it identifies the replay attack, ultimately contributing to more effective protection and detection of covert and replay attacks. The nature of the data is shown as follows:

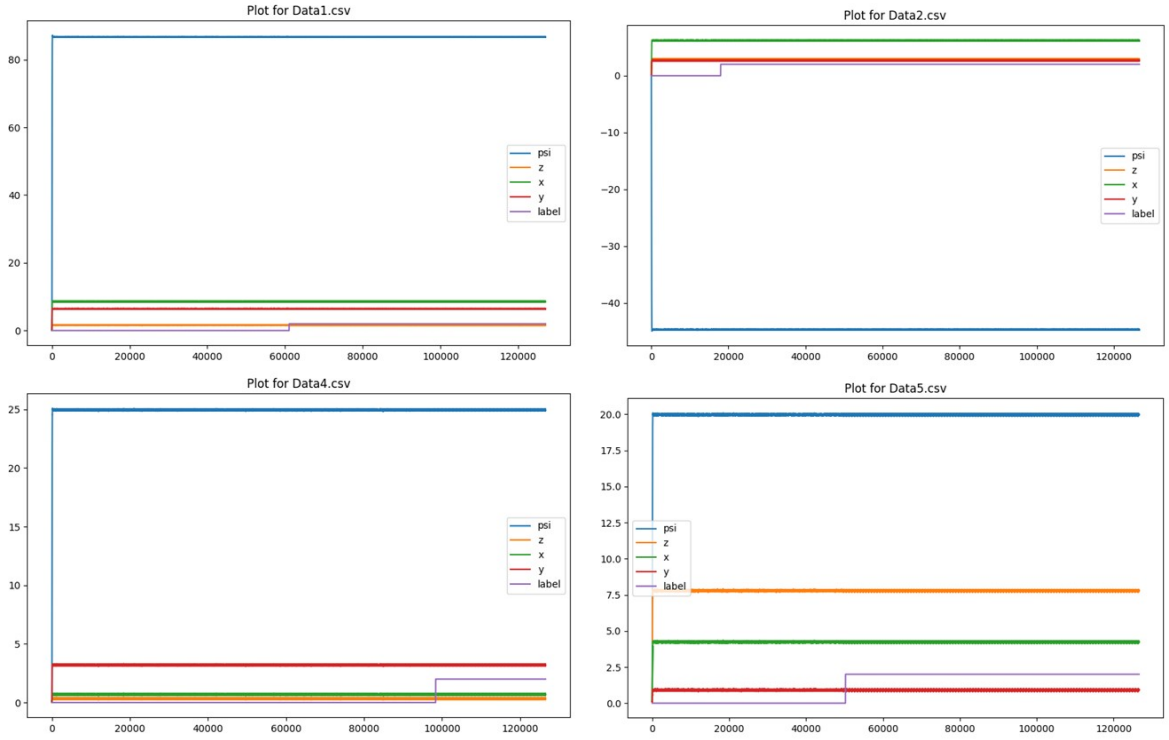

Figure 4.9: Samples of the train data sets face with replay attacks.

Figure 4.9 presents four representative samples of the training datasets used in this research phase, which serve as input to the proposed algorithm. These datasets depict the position outputs of the UAV across different mission scenarios. Although additional datasets have been generated through Monte Carlo simulations, these four samples are specifically selected to exemplify the nature of the algorithm's data processes. As illustrated in the figure, the purple line indicates the label corresponding to the UAV's healthy operation and the periods during which the system is subjected to a replay attack within the training datasets. These examples provide insight into how the algorithm interprets the data to detect and classify the attack.

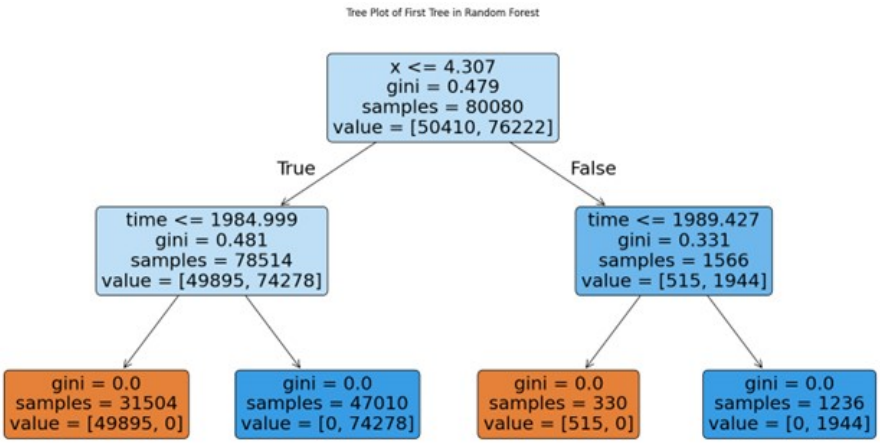

Figure 4.10: Part of GI technique involve in the RF algorithm.

The decision tree in Figure 4.10 illustrates the last part of the feature selection process used by each DT within the Random Forest model for attack detection. The root node is determined by the condition based on the parameter x, where $x \leq 4.307$, and the Gini Index for this node is 0.479, reflecting a significant diversity of data at this point. As the tree progresses through subsequent nodes, decisions are made based on additional parameters, each contributing to the differentiation of the data. A lower Gini Index in the child nodes indicates the increasing power of the selected features in classifying the data, with some nodes achieving a Gini Index of zero, demonstrating complete separation. The decision tree plays a critical role in identifying the most important features for detecting existing attacks, providing a detailed assessment of their relative importance. By using this tree, the model effectively selects key features that improve the performance of the subsequent neural network model. As shown in the figure, the analysis of the Gini impurity provides valuable insights into the features that have the most substantial impact on differentiating between normal operation and attack conditions.

Once the most significant features have been selected through the RF feature selection process, these features are fed into the neural network model for further processing and training. The accuracy and loss validation are shown as follows:

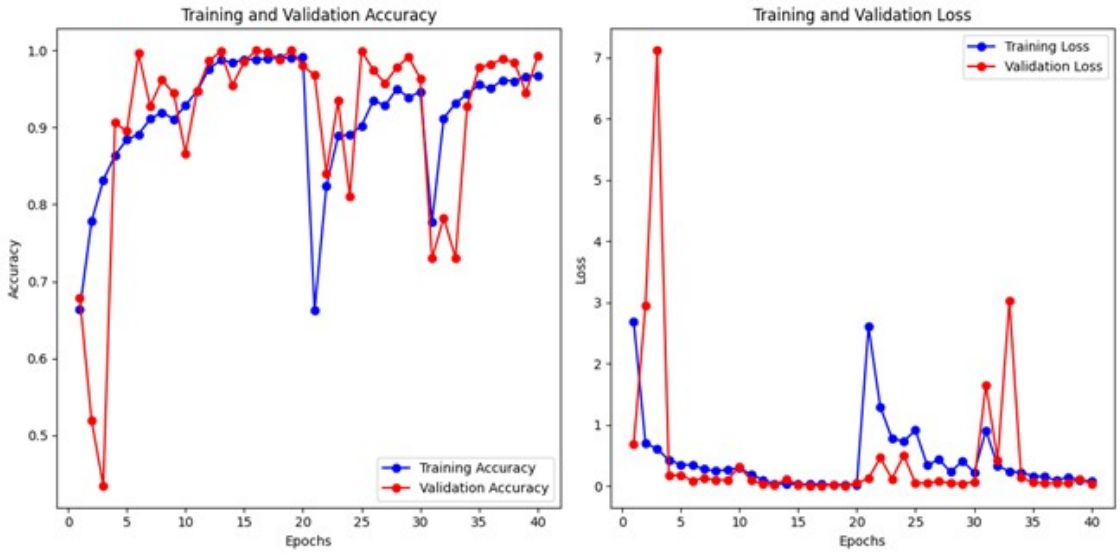

Figure 4.11: The accuracy and loss validation of the model in replay attack scenarios.

Figure 4.11 overviews the ANN model's training and validation performance over 40 epochs. Here, the left graph illustrates training and validation accuracy, while the right graph depicts the corresponding loss values. The general trend in the accuracy graph demonstrates that the model has effectively learned from the training data, with accuracy increasing steadily across epochs. On the other hand, the loss graph shows a rapid decline in both training and validation loss, further suggesting that the model is successfully optimizing its parameters during training. However, certain graph fluctuations highlight areas where the model's generalization could be improved. These variations, particularly noticeable in validation results, suggest potential overfitting, indicating that further refinement of the model's hyperparameters may be necessary to achieve optimal performance.

The designed ANN model, after validation, is now the time for evaluating the test dataset without a label and is used as a test to analyze its performance in predicting replay attacks. In this stage, the focus is on the model's ability to accurately classify the presence of replay attacks based on the input data. The model's effectiveness in detecting these specific attack patterns is evaluated by comparing the actual labels (true attack occurrences) with the predicted labels generated by the model. Analyzing the prediction accuracy allows for identifying potential model limitations and areas requiring further refinement, particularly in distinguishing replay attacks from normal operations, as shown in the figure as follows:

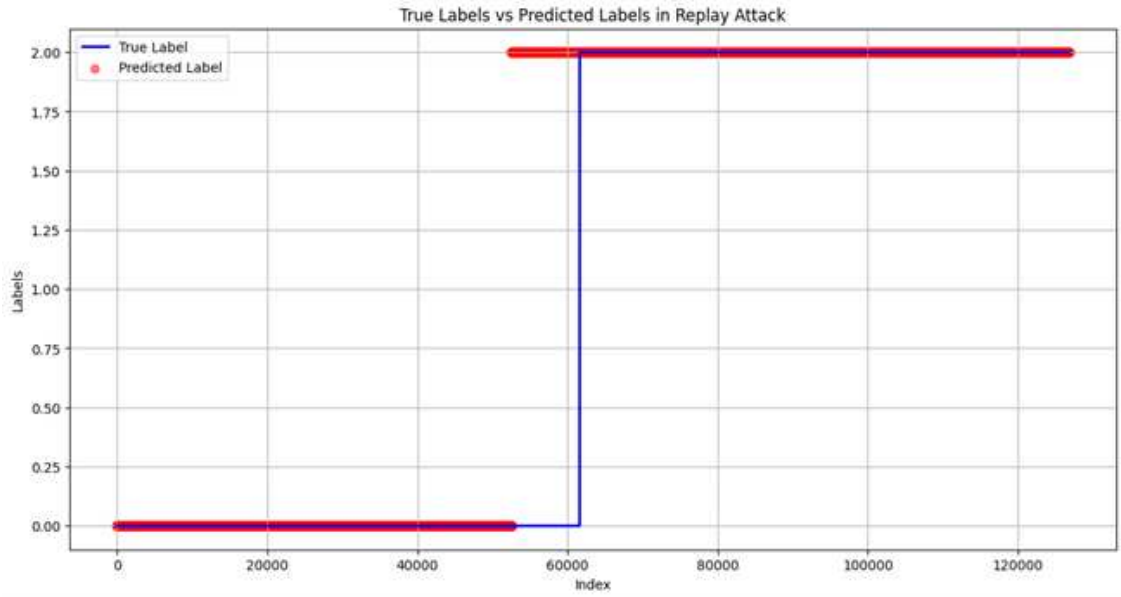

Figure 4.12: Comparison of true label and predicted label.

Figure 4.12 visually compares the actual label and the model's predictions label from the model. The graph clearly shows that in the initial range from 0 to approximately 6000, the actual data labels remain at zero, indicating no Replay attacks, which the model correctly captures. From 6000 to 12000, the labels shift to one, reflecting the presence of replay attacks, and the model accurately predicts these attacks. However, a significant drop in prediction accuracy is observed between 4000 and 6000, where the model fails to identify the attacks. This discrepancy suggests possible model limitations or specific data characteristics that hinder accurate prediction during this period.

The accuracy analysis, such as the AUC and F1 Score, provides a view of the model's discriminate ability and balance between precision and recall. An AUC value of 1 signifies the model's perfect ability to distinguish between attack and non-attack instances across all threshold levels. This reflects optimal true and false positive rates, indicating that the model effectively identifies attacks without confusion between classes. Additionally, the F1 Score of 0.9267 highlights a strong balance between precision and recall, ensuring that the model minimizes false positives while fully capturing all attack instances. The AUC and ROC plot is shown as follows:

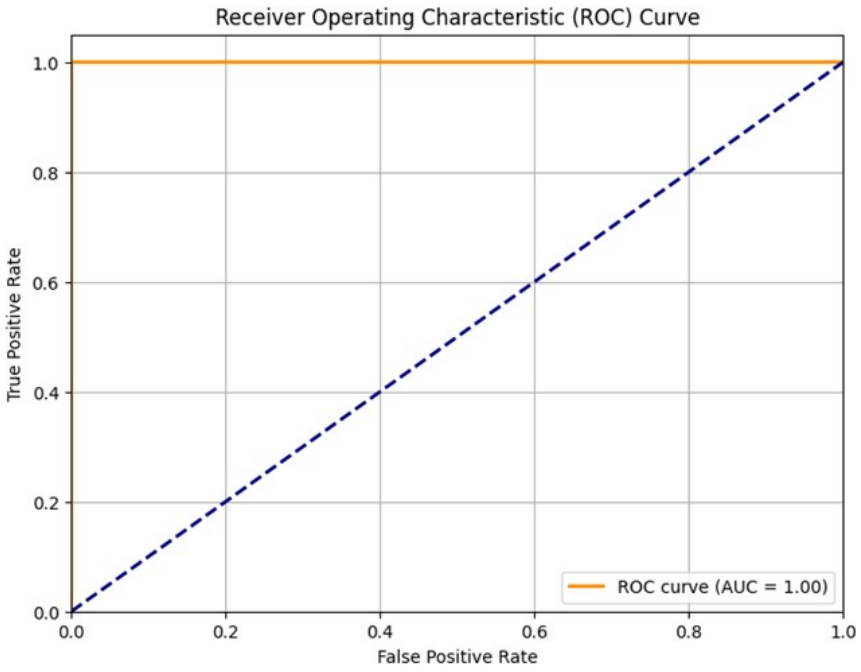

Figure 4.13: AUC and ROC plot.

Figure 4.13 presents the ROC curve and key metrics associated with the model's performance in detecting replay attacks. The ROC curve visually confirms the model's exceptional performance with an AUC of 1, indicating flawless discrimination between attack and non-attack instances. The figure also shows an F1 Score of 0.9267, with a precision of 0.8634 and a recall of 1. These values indicate that %86.34 of the instances identified as attacks were accurate, and all actual attacks were correctly identified without false negatives. This balance between precision and recall reflects the model's capability to maintain high accuracy and effectively reduce errors in detecting replay attacks. Additionally, the confusion matrix shows TF and TN regarding the replay attack detection demonstrated as follows:

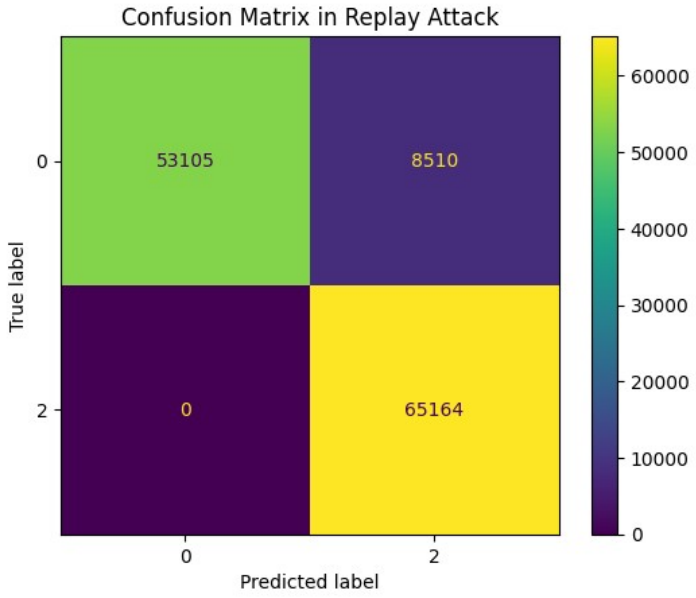

Figure 4.14: Confusion matrix of the model regarding replay attack detection.

Figure 4.14 represents 53105 instances of no attack, and 65164 instances of the attack were correctly identified, while only 8,510 cases were misclassified as attacks. These results confirm the model's capability in accurately detecting replay attacks and highlight the effectiveness of combining decision tree methods with neural networks to enhance detection accuracy.

4.3.3 Covert and Replay Attacks Identification

In this phase of the simulation results, the focus is on evaluating the model's ability to identify different types of attacks, specifically covert and replay attacks, during the UAV's mission. The simulation considers various possible scenarios when two communicated attackers are involved or when a single intelligent attacker gradually gains more knowledge of the system to pose increasingly harmful attacks. These scenarios are critical for understanding the dynamic and evolving nature of attack strategies that adversaries may employ.

The following figure represents the strategy of the data accumulation for the identification problem as follows:

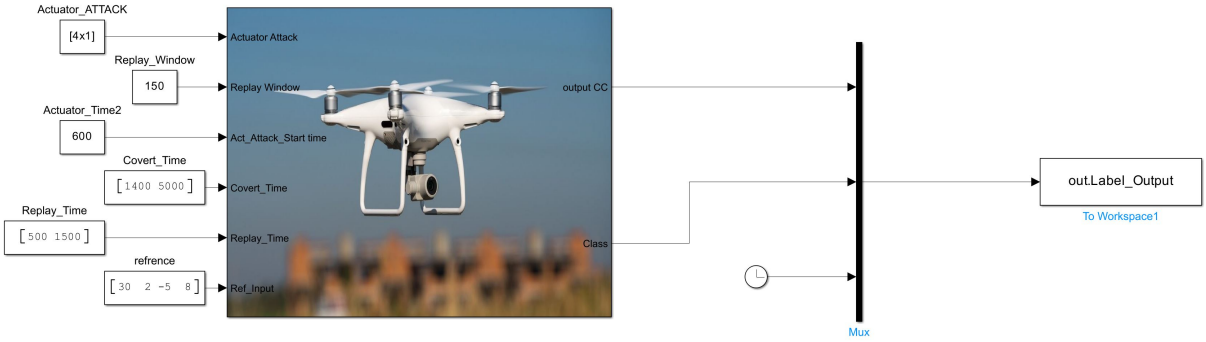

Figure 4.15: Data preparation and training strategies.

Figure 4.15 represents the setup for data extraction from a UAV system, where various parameters such as actuator attack, replay window, and attack times are configured to change the system's functionality. Input data, including reference signals and time windows for covert and replay attacks, are fed into the system to simulate realistic attack scenarios. The simulation output, including the classification results and system positions, is available in C&C as an investigation data set.

The figure helps validate the most likely and logical scenarios an attacker would rely on to execute covert and replay attacks during the mission. By simulating these scenarios, the model's robustness in handling complex attack patterns is assessed, offering a deeper understanding of its potential real-world application in safeguarding against such attacks.

In the following section, five distinct scenarios are evaluated to explore the potential conditions that may arise. Each scenario is carefully analyzed to identify varying complexity and applicability in addressing the problem. The most intricate and feasible scenario is determined by assessing these scenarios, which serves as the primary focus for further analysis. This approach enables a thorough understanding of the problem, concentrating attention on the scenario that presents the most complex and realistic conditions for developing an effective solution.

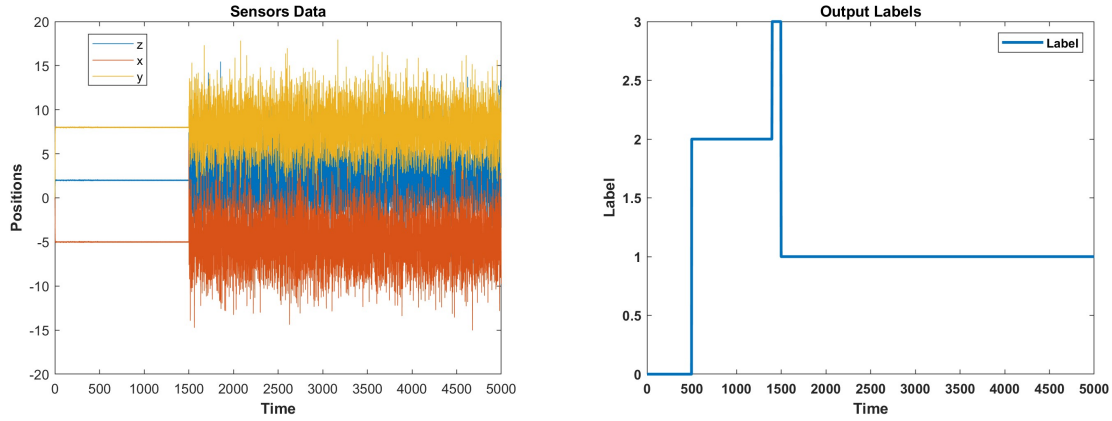

Figure 4.16: $500 < t_r < 1500, 1400 < t_c < 5000$

Figure 4.16 illustrates the replay attack is initiated before the covert attack, with a small overlap period where both attacks exist. This overlap can be seen as a transient phase where the attacker is shifting between different stages of attack injection. This scenario is complex and realistic, as it leverages the transition time to increase the attack's effectiveness while minimizing the detection opportunity. The overlap adds to the complexity, making it challenging for the system to differentiate between the two attack types during this phase, which aligns with sophisticated attack strategies.

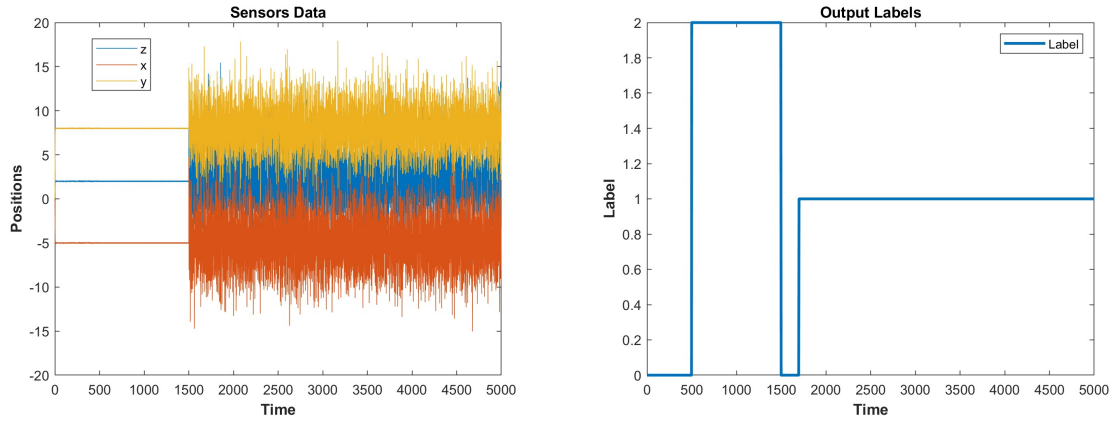

Figure 4.17: $500 < t_r < 1500, 1700 < t_c < 5000$

Figure 4.17 shows a distinct gap between the two types of attacks between 1500 and 1700, where

the replay attack is executed first, followed by the covert attack. While this scenario introduces a level of vulnerability, the gap allows the C&C to detect the UAV's deviation from the expected trajectory. The C&C can thus intervene, potentially preventing further mission progress before the second attack is executed. Due to the gap, this scenario is less advantageous for the attacker, as the C&C may detect the first attack and stop the mission, reducing the likelihood of successful attack progression.

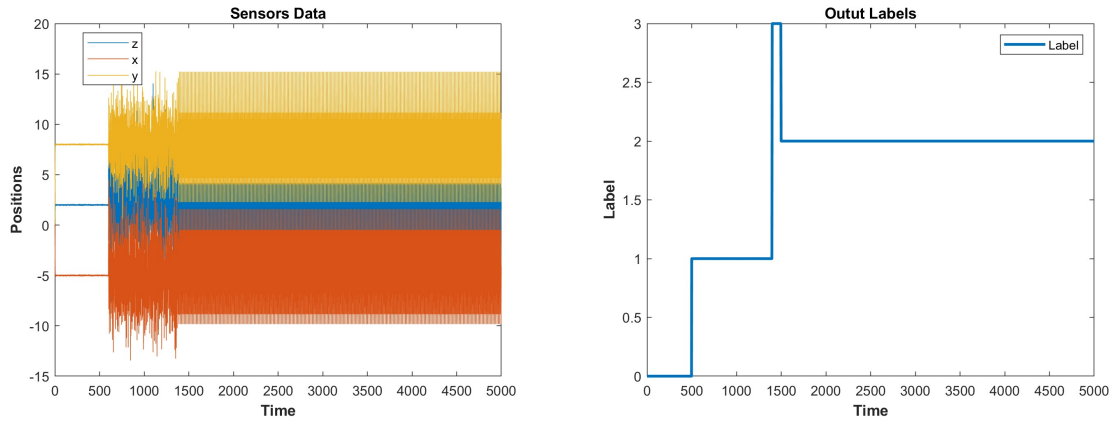

Figure 4.18: $500 < t_c < 15001400 < t_r < 5000$

Figure 4.18 presents where the covert attack is initiated first, followed by a replay attack. Since the replay attack is a subset of the covert attack, when an attacker gains sufficient access to the system's knowledge, the need to downgrade the attack is unnecessary. Having already achieved a covert attack, the attacker would not likely transition to a less sophisticated attack such as a replay. This scenario is, therefore, less plausible, as the covert attack already fulfills the attacker's goal, making the transition to a replay attack redundant and illogical from the perspective of the attack strategy.

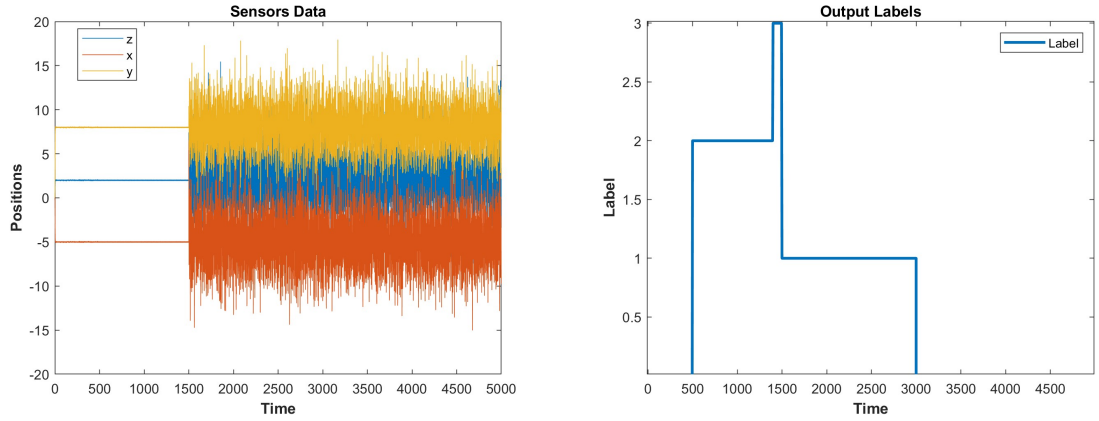

Figure 4.19: $500 < t_r < 1500, 1400 < t_c < 3000$

Figure 4.19 depicts the attacker delaying the attack until the end of the mission. While this strategy allows the attacker to remain undetected for the majority of the mission, it offers limited complexity. The C&C system could easily detect the absence of any attack throughout the mission, reducing the overall challenge for the attacker. This scenario is less effective, as it fails to introduce significant complications for the defence system and does not maximize the attacker's advantage in disrupting the mission.

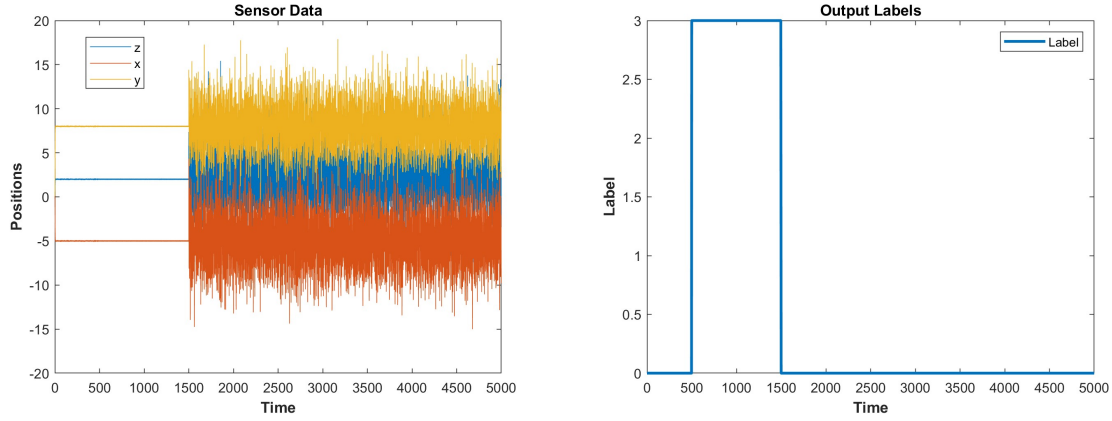

Figure 4.20: $500 < t_r < 1500500 < t_c < 1500$

Figure 4.20 introduces a simultaneous covert and replay attack. Although both attacks are executed

concurrently, the replay attack, being a subset of the covert attack, does not provide additional benefit to the attacker. This setup increases the complexity for the attacker without offering proportional advantages, making it less appealing from a strategic standpoint. The simultaneous execution of both attacks may complicate the attack process without meaningfully enhancing the chances of success.

Among the analyzed scenarios, the first scenario in which the attacker starts by injecting the replay attack and preceding a covert attack with a small overlap emerges as the most complex and strategically sound option. It presents a challenging situation for the C&C system, combining multiple attack phases while minimizing detection opportunities. Therefore, the first scenario is deemed the most plausible and effective, forming the basis for training and testing datasets in this research. On the other hand, an assumption exists that the training scenarios are applied to the datasets by structuring them according to the first scenario. This assumption provides a foundation for the analysis, where the initial scenario is used as a reference framework for organizing and interpreting the training data.

After extracting the data sets, the RF model was employed to identify the key features affecting attack predictions. The structure of the first tree in the Random Forest, showcasing how the model makes decisions based on the selected features, is shown as follows:

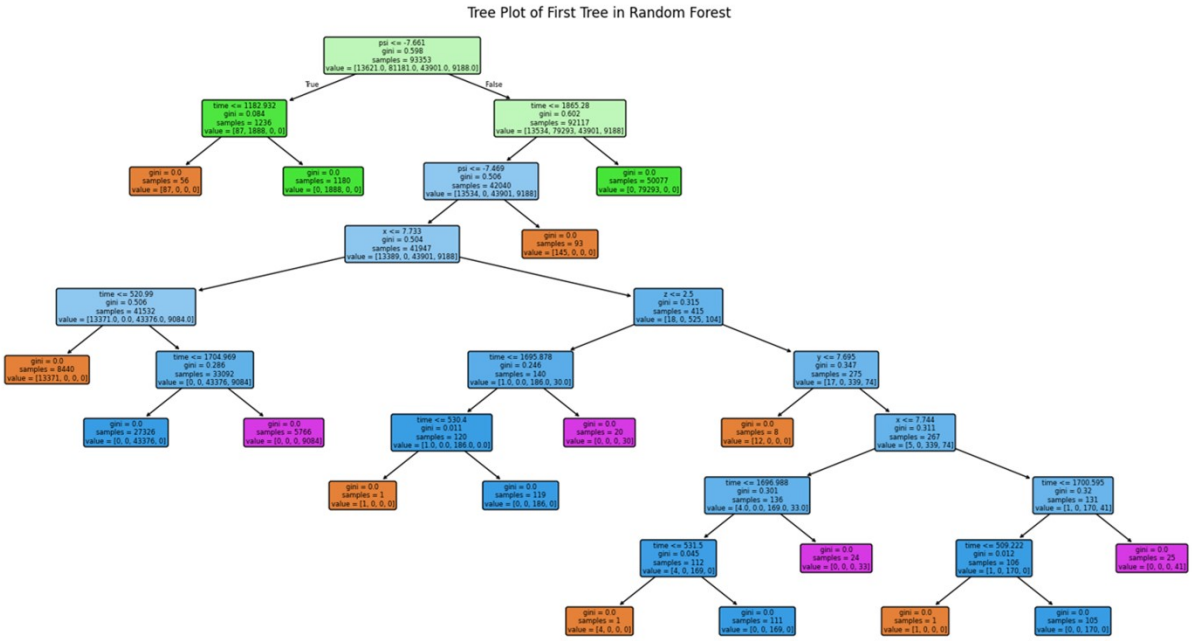

Figure 4.21: Decision tree of the RF model for identifying key features in attack predictions.

Figure 4.21 demonstrates the first tree of RF that features ψ and time are important. The ψ feature, which serves as the primary splitting node, plays a crucial role in differentiating various data samples and directly influences the subsequent pathways of the tree. In other words, this feature has the most significant impact on reducing Gini impurity and accurately classifying the data. The time feature, recognized as the second most important variable, appears at different levels of the decision tree and particularly contributes to branches that differentiate between various attacks. For instance, in the early sections of the decision tree, time is utilized as a determining feature for distinguishing between different attack classes. This characteristic indicates that the timing of events significantly influences the type of attack or abnormal behaviour. The decision tree, through logical partitions based on these two features, assists in accurately identifying and distinguishing between various attack states. Moreover, this tree provides us with critical insights into the patterns present within the data. As the tree branches extend downward, the decision-making model increasingly utilizes relevant features to differentiate classes more accurately. For instance, intermediate nodes that assess features such as ψ and x illustrate how combining these features with ψ and time can facilitate

more precise distinctions between attack and benign classes. Furthermore, each colour of the decision tree nodes represents the Gini impurity and the number of samples per class in each node. Generally, nodes with lighter colours indicate higher purity, suggesting that a larger proportion of samples belong to a specific class in these nodes. This finding confirms that the model effectively utilizes the selected features for classifying samples.

Regarding training and validation, the neural network model results after feature selection by RF. The accuracy and error graphs for different datasets are shown as follows:

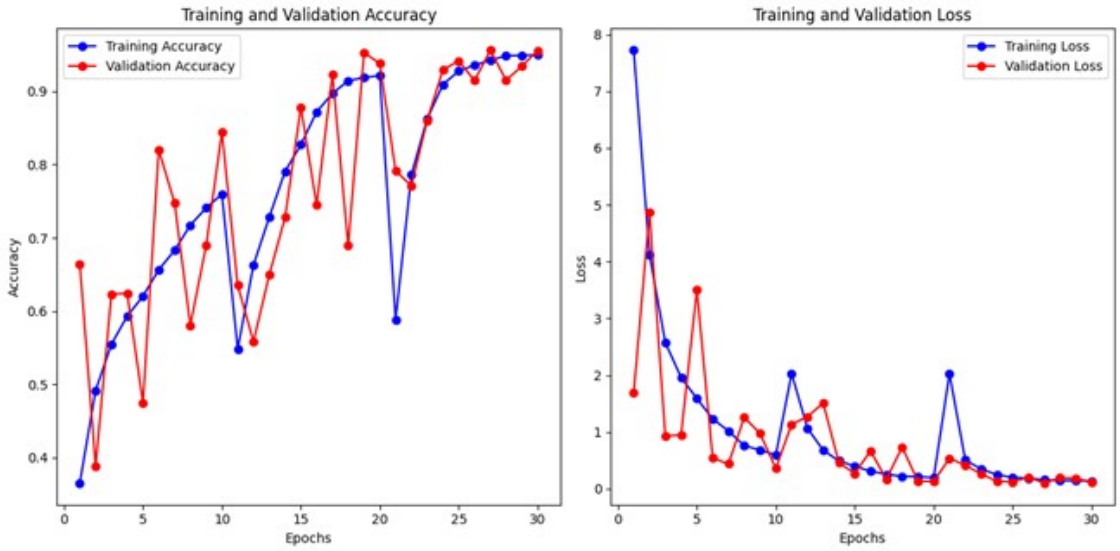

Figure 4.22: Comparison of training and validation accuracy and error rates of the neural network model across different datasets. (Left sub-figure: Comparison of training and validation accuracy over training epochs. Right sub-figure: Comparison of training and validation error rates over training epochs.)

In the left graph of Figure 4.22, the model's accuracy during various training epochs is displayed, the training set (blue) and the validation set (red). The overall trend in the graphs indicates that the model's accuracy consistently rises as the number of epochs increases. Initially, this increase in accuracy is accompanied by greater fluctuations; however, after approximately 20 epochs, the accuracy stabilizes and approaches a high value near 95%. This reflects the model's ability to learn and generalize on unseen data. Although the validation accuracy experiences some instability in certain instances, the overall trend shows improvement. The right graph of Figure 4.22 depicts the

training and validation error rates over the epochs. At the beginning of the training, the error rate is relatively high but gradually decreases over time. This decline is particularly rapid during the first five epochs, after which it adopts a more gradual trend. The significant reduction in error during the early epochs indicates the model's swift learning of patterns in the training data. The error approaches a lower value in the later epochs, suggesting the model's stability and consistency.

In the performance analysis of the model using the ROC curve for attack identification, the AUC for each class is calculated separately to evaluate the model's ability to detect different types of attacks. This matter is shown visually as follows:

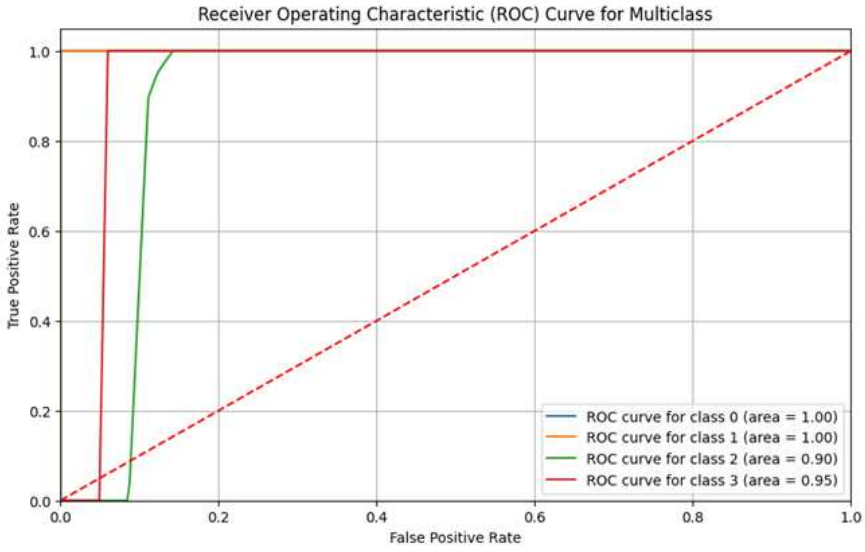

Figure 4.23: ROC curve and AUC comparison for different classes in attack detection.

Figure 4.23 demonstrates the accuracy of classes; the ROC curves for 0 and 1 (Blue and Orange Lines) classes are significantly inclined towards the upper left corner. This indicates that the model has effectively identified these classes. An AUC value of 1.00 for both classes suggests that the model has operated without any false positives and accurately detects these classes. However, the ROC curve for class 2 (Green Line) is positioned somewhat farther from the upper left corner, indicating that the model does not perform as well in detecting this class compared to Classes 0 and 1. The AUC value of 0.90 indicates that the model still demonstrates a reasonable level of performance in distinguishing Class 2, although some false positives are likely. The ROC curve for Class 3 (Red Line) leans significantly toward the diagonal center line, suggesting weaker performance than the

other classes. However, the AUC value of 0.95 shows that the model has still identified this class with reasonable accuracy despite a relatively higher incidence of false positives.

The predicted and actual label regarding the test data set is shown as follows:

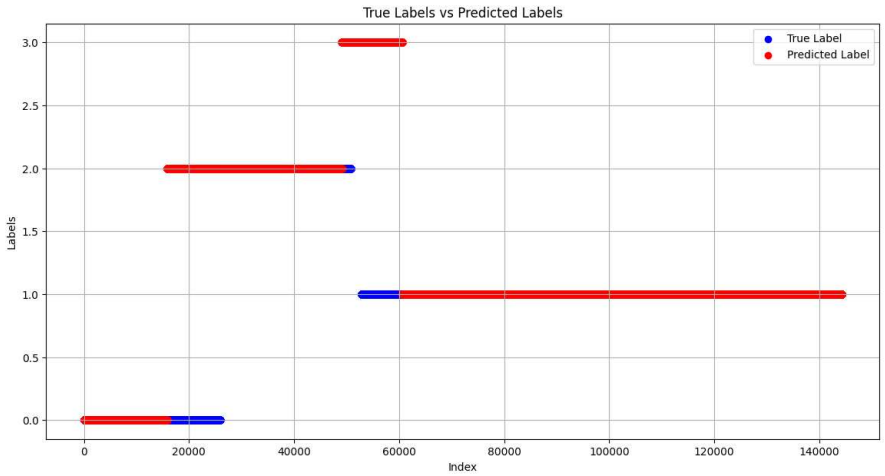

Figure 4.24: Comparison of true and predicted labels

Figure 4.24, demonstrates the model is able to predict most of the existing labels correctly; however, the confusion matrix for the attack identification model illustrates the model's performance for each point as follows:

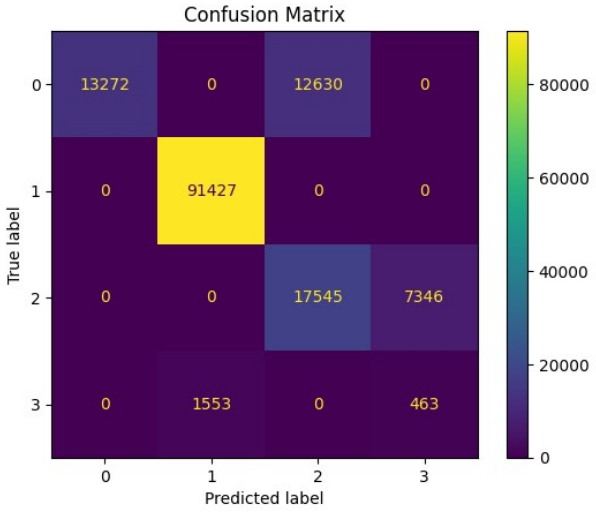

Figure 4.25: Identification confusion matrix

Figure 4.25 represents that class 0 has been identified in this matrix with high accuracy, with 12,630 out of 13,272 actual samples correctly identified and no samples misclassified into other classes. Additionally, class 1 has demonstrated remarkable success by correctly identifying all 91,427 samples without any false positives or negatives. This optimal performance in recognizing these two classes indicates the model's strong capability in differentiating and identifying these categories. In contrast, the results for classes 2 and 3 present more significant challenges for the model. For Class 2, the model correctly identified 17,545 out of 24,891 actual samples, but 7,346 samples were misclassified, indicating the presence of false positives and false negatives in this section. Furthermore, the performance for class 3 is notably poor; from 2,016 actual samples, only 463 were correctly identified, while 1,553 samples were mistakenly assigned to other classes. These results highlight the need for optimization and improvement in identifying the scenario in which both attacks are involved simultaneously.

4.3.4 Performance Analysis with Alternative Algorithms

Regarding evaluating and comparing existing work by literature, [11] propose a deep learning framework designed to detect, diagnose, and localize covert attacks within networked industrial control systems (ICS). Their approach integrates an autoencoder for feature extraction, a recurrent neural network (RNN) to model system behaviour under normal conditions, and a deep neural network (DNN) classifier for detecting and distinguishing between cyber-attacks and natural faults. This framework investigates spatial and temporal data from ICS networks, effectively capturing the complex interactions among subsystems to reduce false alarms and pinpoint the origin of attacks. Similar to existing research, this approach focuses on identifying covert attacks due to their undetectable nature. Using a data-driven algorithm aligns with the proposed methodology, aiming to improve the reliability and robustness of CPSs through machine learning, making it comparable to the suggested algorithm and results. The algorithm proposed in [11] is applied to an existing dataset, allowing for a direct comparison with the algorithm developed in this research.

Multiple datasets were processed using a deep learning approach to evaluate model performance for classification tasks. Models were trained on data from different files, with each training session running over ten epochs. Initial warnings indicated that input shapes should not be explicitly defined in

layers when using the Sequential model, suggesting an adjustment for best practices. Model performance accuracy for the trained datasets achieved a maximum of 61.2%. In evaluating classification models, accuracy and confusion matrices are fundamental metrics for assessing performance. The accuracy and loss of their algorithm on the existing data set are shown as follows:

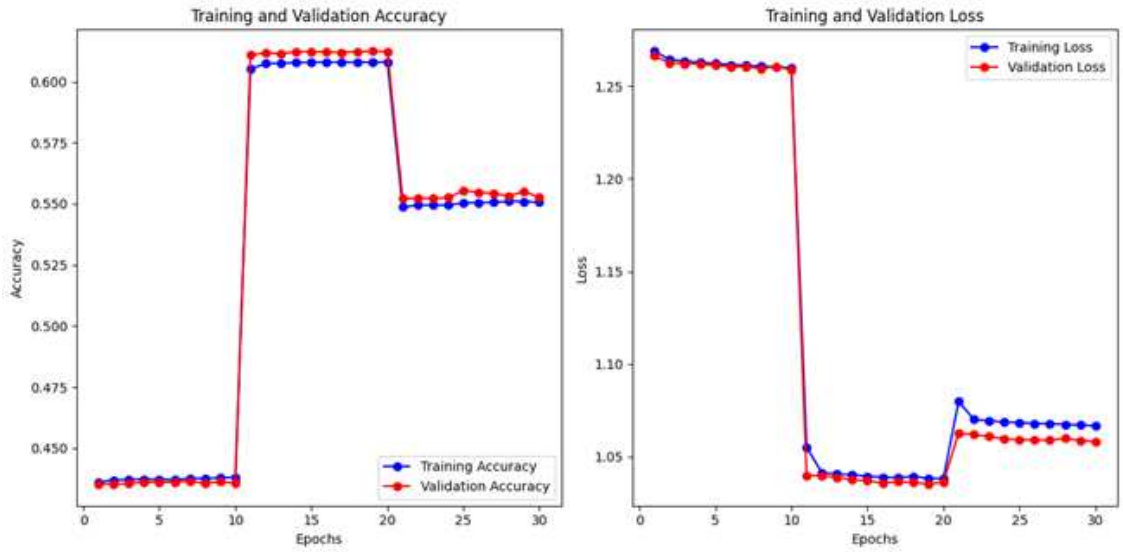

Figure 4.26: Comparison of training and validation accuracy and error rates of the neural network model [11] algorithm across different datasets. (Left sub-figure: Comparison of training and validation accuracy over training epochs. Right sub-figure: Comparison of training and validation error rates over training epochs.)

Figure 4.26 in comparison to Figures 4.5, 4.11, and 4.22 demonstrates less accuracy and more loss. However, the other validation of paper [11] is shown as follows:

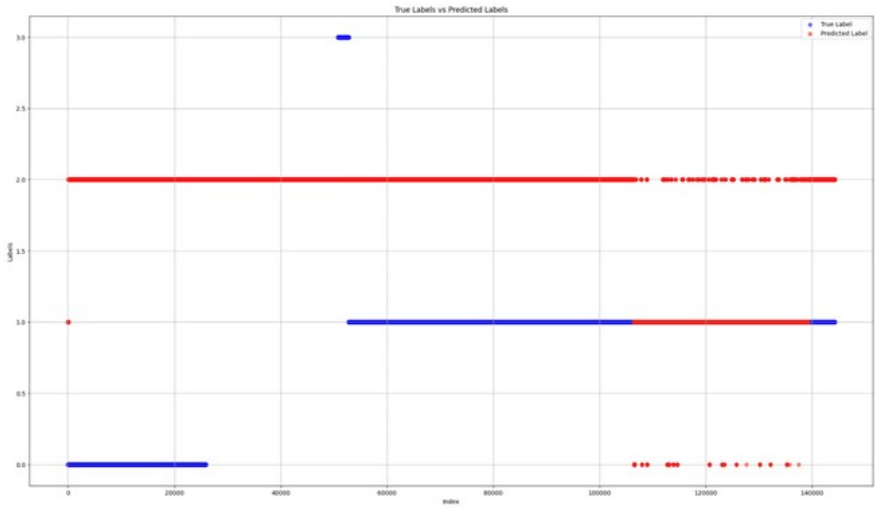

Figure 4.27: Comparison of true label and predicted label with respect to [11] algorithm.

Figure 4.27 represents the ratio of correctly predicted instances to the total number of instances and provides a straightforward measure of overall model effectiveness. However, relying solely on accuracy can be misleading, especially in imbalanced datasets where one class significantly outnumbers others. This matter is clearly shown in the confusion matrix as follows:

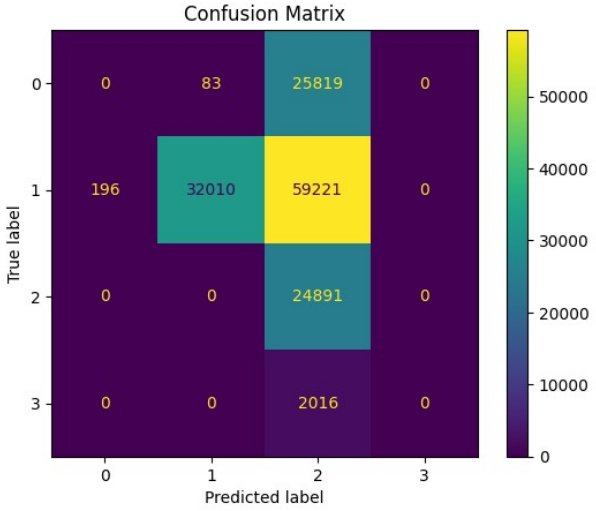

Figure 4.28: Identification confusion matrix with respect to [11] algorithm.

The confusion matrix complements accuracy by providing a more detailed breakdown of the model's

performance across all classes. It presents counts of true positives, true negatives, and false positives for each class, allowing for a nuanced analysis of where the model succeeds and fails. From the confusion matrix and the calculated accuracy, the model demonstrates a significant ability to classify instances of label 1 (with a true positive count of 32,010). This means the algorithm can effectively identify the covert attack. However, many misclassifications for label 0 (25,819 times predicted as label 2) indicate a substantial issue in recognizing healthy and replay attack classes.
The ROC curves for each class reveal distinct characteristics regarding model performance shown as follows:

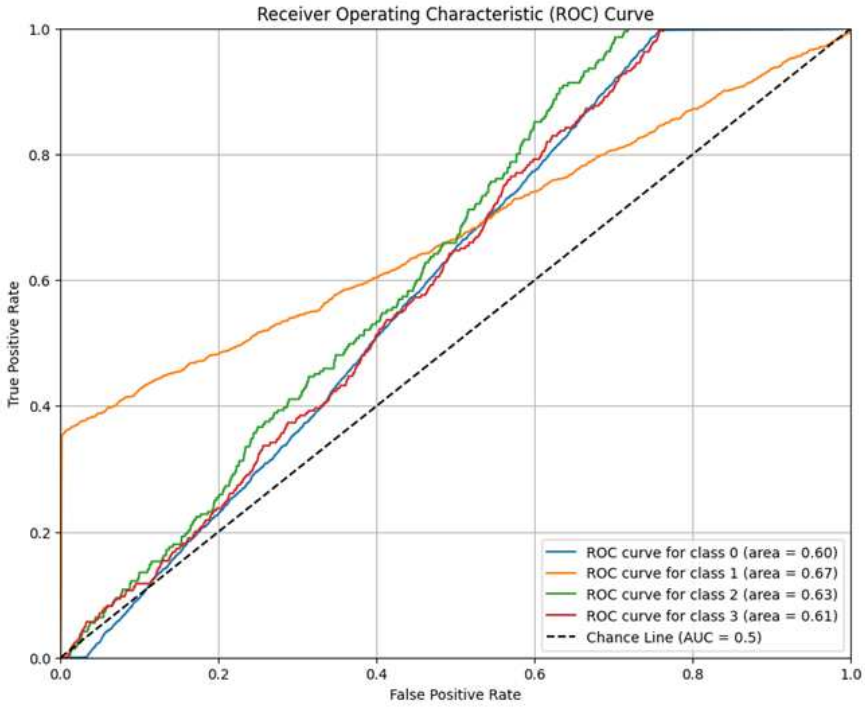

Figure 4.29: ROC curve comparison for different classes in attack identification with respect to [11] algorithm.

Here, The ROC for Class 1, with an AUC of 0.67, signifies a moderate ability of the model to differentiate between positive and negative instances for this class. The AUC for Class 2 is 0.63, slightly lower than Class 1. This implies that while the model maintains reasonable discriminative power, improvement is potential. Class 0 and Class 3 both exhibit lower AUC values of 0.60 and 0.61, respectively. These results indicate a weaker performance in discriminating these classes from

others, suggesting that the model struggles to identify positive instances accurately.

The classification report provides a detailed overview of the model's performance across different classes, quantifying its effectiveness through precision, recall, and F1-score metrics as follows:

Class	Precision	Recall	F1-Score
0	0	0	0
1	1	0.35	0.52
2	0.22	1	0.36
3	0	0	0
Accuracy		**0.39**	
Macro Avg	0.3	0.34	0.22
Weighted Avg	0.67	0.39	0.39

Table 4.1: Classification report with respect to [11] algorithm.

Table 4.1 demonstrates the model performance across the four classes varies notably, revealing key strengths and limitations. For Class 1, the precision reaches 1.00, indicating perfect prediction accuracy for identified instances, though the recall is only 0.35. This leads to a moderate F1-score of 0.52, pointing to a need for improved recall. Conversely, Class 2 achieves a recall of 1.00 but a low precision of 0.22, suggesting high false positives despite capturing all actual instances. Classes 0 and 3 report zero precision and recall, indicating a complete lack of accurate predictions for these classes. The overall accuracy of the model is 0.39. Recall accuracy measures a model's ability to correctly identify all relevant instances, calculated as the ratio of true positives to the sum of true positives and false negatives. which underlines the need for refinement to improve the effectiveness of classification, particularly for poorly performing classes.

4.3.5 Comparison Suggested with Alternative Algorithm

The [11] proposed a sophisticated hybrid framework for attack identification. It begins with an autoencoder for feature extraction, reducing the dimensionality of high-dimensional sensor data. An LSTM-RNN follows this for capturing temporal dependencies in the data, as these models excel in modelling time-series data. To improve the robustness of attack detection, residual calculation

using RNNs is employed to assess deviations from expected patterns. This residual information is passed to DNN for final attack identification. LSTM-RNN, in this research, helps handle non-linear temporal predictions, making it particularly suited for time-dependent systems. The model operates in multiple stages to ensure the features and temporal information are adequately captured before final attack identification. In contrast, the suggested algorithm takes a more straightforward approach using a Random Forest model for feature selection. This focuses on identifying the most important features from the dataset, which helps simplify the neural network's task by reducing input complexity. A simple neural network with dense layers is employed for classification, and SMOTE (Synthetic Minority Over-sampling Technique) is used to balance the class distribution. The neural network is trained using the Adam optimizer, with accuracy being the key metric for model evaluation. This methodology provides a more straightforward and effective model for attack identification, especially when dealing with imbalanced datasets, such as in quadcopters. When comparing the two methods, the first approach leverages advanced techniques like autoencoders and LSTM-RNNs to handle both feature extraction and temporal dependencies. This allows for more detailed sensor data analysis, especially in systems with crucial time components. In contrast, the second model simplifies feature selection using Random Forests and focuses primarily on classification, balancing the dataset through SMOTE. The Random Forest technique, while effective in feature importance ranking, may not capture temporal dependencies as effectively as LSTM-based models. However, the simpler neural network architecture in the second approach may result in a faster and less computationally expensive solution. The table demonstrates this comparison visually as follows:

Feature/Model	Algorithm [11]	Suggested Algorithm
Feature Extraction	Autoencoder for dimensionality reduction of sensor data	Random Forest for importance feature selection
Temporal Dependency	LSTM-RNN for temporal data	No explicit temporal modeling required
Attack Identification	DNN using residual-based	Simple NN with dense layers
Imbalance Handling	Not specified	SMOTE for imbalance classes
Optimization	Not specified	Adam optimizer
System Focus	Time-sensitive systems	Not require time-sensitive systems

Table 4.2: Comparison of suggested and [11] algorithms.

Table 4.2 effectively compares the proposed algorithm and relevant literature. The approach in [11] demonstrates effectiveness primarily for temporal data, a significant limitation as it restricts applicability across various scenarios. In contrast, the proposed algorithm emphasizes accurate feature selection before the NN model, which helps suggest a less complex model. This focused approach conserves computational resources and simplifies the model, resulting in more efficient processing compared to the method in [11].

Regarding performance comparison, [11] achieves lower accuracy with only 39% accuracy overall. The precision, recall, and F1-scores for class 0 and 3 are particularly low, indicating that the model struggles with some classes. It also has a macro average precision of 0.30, which suggests the model's predictions are not robust across different classes. Despite the complex architecture, the model does not seem to handle attack identification on existing data sets effectively, as it fails to accurately classify certain classes. However, the suggested algorithm in this research achieves an overall accuracy of 85%. The results show significantly higher performance for most classes, especially class 1, with an F1-score of 0.99. This algorithm is better suited for handling attack detection and identification problems. The macro average F1-score of 0.6, much higher than the [11], implies that the suggested model performs more consistently across different attack types. Table 4.3.5 shows the performance comparison as follows:

Class	Existing Literature [11]			Suggested Algorithm		
	Precision	Recall	F1-Score	Precision	Recall	F1-Score
0	0	0	0	1	0.51	0.68
1	1	0.35	0.52	0.98	1	0.99
2	0.22	1	0.36	0.58	0.7	0.64
3	0	0	0	0.06	0.23	0.09
Accuracy		**0.39**			**0.85**	
Macro Avg	0.3	0.34	0.22	0.66	0.61	0.6
Weighted Avg	0.67	0.39	0.39	0.9	0.85	0.86

Table 4.3: Performance Comparison between Existing Literature and Suggested Algorithm

Concerning the comparison, while Reference [11] utilizes a sophisticated hybrid framework combining LSTM, autoencoders, and DNN, its performance in detecting attacks is subpar, likely due to its inability to handle imbalanced datasets and certain attack types. However, the suggested model is more straightforward and performs better overall by focusing on feature selection through Random Forest and employing SMOTE to handle data imbalance. Using a simple neural network optimized with Adam further contributes to more effective attack detection in quadcopter systems.

4.4 Conlusion and Future work

This research has addressed covert and replay attack detection and identification within CPSs using a data-driven algorithm, achieving an accuracy of 85% in identifying attacks. The model also demonstrated the capability to detect transition periods where an attack is updating itself, thus posing a more sophisticated attack like covert. Emphasis was placed on using RF for feature selection, effectively reducing computational procedures and resource demands for the NN model. While the achieved accuracy is acceptable for initial evaluations, further enhancements are needed to ensure suitability for real-time systems that demand higher precision.

Future work should focus on testing this algorithm against additional attacks that exist in literature, such as zero dynamic attacks, to assess its resilience in multi-attack scenarios during operation. Moreover, although the model performed well under noisy conditions, it would benefit from further evaluation in the presence of faults and disturbances to ensure robust performance. These refinements would strengthen the algorithm's applicability to safety-critical systems, such as drones, where high reliability and security to various environmental and operational challenges are essential.

Chapter 5

Conclusion and Future Works

This chapter briefly summarizes the thesis outcomes, and some future work directions are mentioned.

5.1 Conclusions

This thesis addresses the challenge of detecting stealthy cyberattacks, particularly covert and replay attacks, within model-based and data-driven frameworks. In developing an effective detection methodology, CPS security and system integrity have been central considerations, ensuring that the added matrix to the system not only enhances the system's detection mechanisms but also boosts the system's security without causing instability. The presented procedures contribute to the security of CPS, helping to prevent or delay system exploitation by stealthy attacks.

This thesis begins by exploring the literature to identify current research directions and pinpoint existing gaps, which informed the development of the problem statement outlined in Chapter 1. Focusing on critical systems vulnerable to cyberattacks, a nonlinear quadrotor is selected as a representative case study. The mathematical modelling of this UAV, including linearization steps and its integration within the CPS framework, is detailed. Chapter 2 presents key command and control components, such as the LQI controller, Kalman filter observer, and the detector's mathematical formulations. Additionally, relevant literature on model-based and data-driven approaches to covert attack detection is reviewed, establishing a solid theoretical foundation for the primary contributions

of this thesis.

Chapter 3 presents the initial contributions of this thesis within a model-based framework, focusing on the development of a coding matrix for an augmented linear system. This matrix is designed to maximize the impact of an attacker's input on the residual signal while simultaneously minimizing its effect on the system itself. Additionally, security concerns for critical systems are addressed through the implementation of periodic coding designs that prioritize system integrity. The dictionary design is also explored, bridging theoretical contributions with practical applications to reflect real-world scenarios. This approach not only effectively detects covert attacks but is also capable of identifying replay attacks, further enhancing system security.

Chapter 4 details contributions made within the data-driven framework, which is particularly suited for high-fidelity systems where deriving a complete mathematical model is challenging and only partial system data is available. This chapter centers on the development of an optimized neural network model designed to achieve reliable accuracy in detecting covert and replay attacks. To enhance the model's effectiveness, feature selection is identified as the most appropriate data preparation step, leading to the implementation of a random forest algorithm. This algorithm selects significant features from the available data, serving as inputs to the neural network for the classification of attack types. The developed approach achieves commendable accuracy, exceeding expectations by not only detecting covert and replay attacks but also successfully distinguishing and labelling each type, enhancing the system's robustness against cyber threats.

Simulation results are presented to demonstrate the proposed solution's effectiveness concerning each contribution.

5.2 Future Works

For future work, several enhancements could complete the current contributions. Within the model-based framework, security techniques could be introduced to ensure the integrity of the pre-defined dictionary and secure the transmission of labels through a protected channel, enhancing the security of the existing design. Additionally, while the proposed coding design enables the detection of covert and replay attacks, future research could focus on developing methods for mitigation and

compensation following attack detection. Also, the proposed solutions could be evaluated in the scenarios in which the fault and disturbances are considered in the mathematical representation.

In the data-driven framework, there is potential to improve the algorithm's accuracy, particularly in cases where multiple attacks occur simultaneously. While the neural network demonstrates high accuracy in testing and validation with current datasets, future work could explore methods that retain all candidate models rather than selecting only one, allowing decision-making to leverage both pre-defined models and the unique characteristics of the test dataset. Further adjustments to the random forest hyperparameters could enable feature sorting rather than selecting only the most significant features, yielding a potentially substantial improvement in detection capabilities. Finally, the algorithm could be extended and tested on a broader range of attack types with varied characteristics, enhancing its adaptability and accuracy across diverse scenarios.

Bibliography

[1] K. B. Adedeji and Y. Hamam, "Cyber-physical systems for water supply network management: Basics, challenges, and roadmap," *Sustainability*, vol. 12, no. 22, p. 9555, 2020.

[2] F. Devoti, P. Mursia, V. Sciancalepore, and X. Costa-Pérez, "Taming aerial communication with flight-assisted smart surfaces in the 6g era: Novel use cases, requirements, and solutions," *IEEE Vehicular Technology Magazine*, 2023.

[3] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "A secure control framework for resource-limited adversaries," *Automatica*, vol. 51, pp. 135–148, 2015.

[4] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, 2018.

[5] "6dof motion platform." https://rush-racing.com/?product=6dof-motion-platform. Accessed: 2024-10-10.

[6] Shubham, "Kalman filter explained," 2020. Accessed: 2023-02-13.

[7] A. Hoehn and P. Zhang, "Detection of covert attacks and zero dynamics attacks in cyber-physical systems," in *2016 American Control Conference (ACC)*, pp. 302–307, IEEE, 2016.

[8] Z. Niu, G. Zhong, and H. Yu, "A review on the attention mechanism of deep learning," *Neurocomputing*, vol. 452, pp. 48–62, 2021.

[9] J. Zhang, Y. Hao, R. Fan, and Z. Wang, "An ultra-short-term pv power forecasting method for changeable weather based on clustering and signal decomposition, energies 16 (7)(2023) 3092."

[10] A. G. Lazcano-Herrera, R. Q. Fuentes-Aguilar, I. Chairez, L. M. Alonso-Valerdi, M. Gonzalez-Mendoza, and M. Alfaro-Ponce, "Review on bci virtual rehabilitation and remote technology based on eeg for assistive devices," *Applied Sciences*, vol. 12, no. 23, p. 12253, 2022.

[11] D. Li, P. Ramanan, N. Gebraeel, and K. Paynabar, "Deep learning based covert attack identification for industrial control systems," in *2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 438–445, IEEE, 2020.

[12] E. A. Lee, "Cyber physical systems: Design challenges," in *2008 11th IEEE international symposium on object and component-oriented real-time distributed computing (ISORC)*, pp. 363–369, IEEE, 2008.

[13] C. C. John, "Challenges in the design of cyber-physical systems," *Revista Ingenierías US-BMed*, vol. 1, no. 1, pp. 6–14, 2010.

[14] K. Yoon, D. Park, Y. Yim, K. Kim, S. K. Yang, and M. Robinson, "Security authentication system using encrypted channel on uav network," in *2017 First IEEE International Conference on Robotic Computing (IRC)*, pp. 393–398, IEEE, 2017.

[15] V. Gunes, S. Peter, T. Givargis, and F. Vahid, "A survey on concepts, applications, and challenges in cyber-physical systems.," *KSII Trans. Internet Inf. Syst.*, vol. 8, no. 12, pp. 4242–4268, 2014.

[16] A. Guo, D. Yu, Y. Hu, S. Wang, T. An, and T. Zhang, "Design and implementation of data collection system based on cps model," in *2015 International Conference on Computer Science and Mechanical Automation (CSMA)*, pp. 139–143, IEEE, 2015.

[17] B. Matt, "Computer security: art and science," 2002.

[18] L. Watkins, J. Ramos, G. Snow, J. Vallejo, W. H. Robinson, A. D. Rubin, J. Ciocco, F. Jedrzejewski, J. Liu, and C. Li, "Exploiting multi-vendor vulnerabilities as back-doors to counter the threat of rogue small unmanned aerial systems," in *Proceedings of the 1st ACM MobiHoc workshop on mobile IoT sensing, security, and privacy*, pp. 1–6, 2018.

[19] D. Rudo and D. K. Zeng, "Consumer uav cybersecurity vulnerability assessment using fuzzing tests," *arXiv preprint arXiv:2008.03621*, 2020.

[20] A. Shafique, A. Mehmood, and M. Elhadef, "Survey of security protocols and vulnerabilities in unmanned aerial vehicles," *IEEE Access*, vol. 9, pp. 46927–46948, 2021.

[21] M. A. Rahman, M. T. Rahman, M. Kisacikoglu, and K. Akkaya, "Intrusion detection systems-enabled power electronics for unmanned aerial vehicles," in *2020 IEEE CyberPELS (CyberPELS)*, pp. 1–5, IEEE, 2020.

[22] Y. Mo, T. H.-J. Kim, K. Brancik, D. Dickinson, H. Lee, A. Perrig, and B. Sinopoli, "Cyber–physical security of a smart grid infrastructure," *Proceedings of the IEEE*, vol. 100, no. 1, pp. 195–209, 2011.

[23] Y. Zhang, Y. Liu, and H. Chen, "Security and privacy in smart grids: Challenges and opportunities," *IEEE Transactions on Industrial Informatics*, vol. 13, no. 4, pp. 1905–1915, 2017.

[24] J. Smith, A. Brown, and M. Johnson, "Mitigating denial-of-service attacks in cyber-physical systems," *Journal of Cybersecurity Research*, vol. 5, no. 2, pp. 112–130, 2019.

[25] S. Lee and J. Kim, "Adaptive security measures for mitigating denial-of-service attacks in cyber-physical systems," *International Journal of Critical Infrastructure Protection*, vol. 21, pp. 45–58, 2018.

[26] C. Xu, Z. Wang, and P. Zhang, "Security and privacy in cooperative intelligent transportation systems: Challenges and solutions," *IEEE Transactions on Intelligent Transportation Systems*, vol. 20, no. 2, pp. 709–723, 2019.

[27] N. Sastry, S. Shankar Sastry, and D. Wagner, "Secure verification of location claims," *ACM Transactions on Information and System Security*, vol. 7, no. 4, pp. 499–522, 2004.

[28] Y. Liu, J. Wu, and A. Smith, "Towards resilient cyber-physical systems: A comprehensive review of replay attack mitigation techniques," *Journal of Cybersecurity Engineering*, vol. 12, no. 3, pp. 211–228, 2016.

[29] H. Chen, L. Zhang, and Q. Wang, "A comparative analysis of cryptographic approaches for replay attack prevention in cyber-physical systems," *IEEE Transactions on Dependable and Secure Computing*, vol. 15, no. 5, pp. 756–769, 2018.

[30] R. Gupta, S. Patel, and M. Johnson, "Temporal logic-based approaches for timestamp verification in cyber-physical systems," *International Journal of Security and Privacy*, vol. 8, no. 2, pp. 45–62, 2020.

[31] N. Babadi and A. Doustmohammadi, "A moving target defence approach for detecting deception attacks on cyber-physical systems," *Computers and Electrical Engineering*, vol. 100, p. 107931, 2022.

[32] C. Lei, H.-Q. Zhang, J.-L. Tan, Y.-C. Zhang, and X.-H. Liu, "Moving target defense techniques: A survey," *Security and Communication Networks*, vol. 2018, no. 1, p. 3759626, 2018.

[33] J. Li, J. Wu, and R. Lu, "Cyber-physical covert attacks and defenses in uav systems," *IEEE Transactions on Vehicular Technology*, vol. 67, no. 11, pp. 10941–10954, 2018.

[34] Y. Liu and J. Wu, "Covert attacks in cyber-physical systems: A comprehensive overview," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 2, pp. 785–797, 2020.

[35] M. Taheri, K. Khorasani, I. Shames, and N. Meskin, "Data-driven covert-attack strategies and countermeasures for cyber-physical systems," in *2021 60th IEEE Conference on Decision and Control (CDC)*, pp. 4170–4175, IEEE, 2021.

[36] K. Gheitasi and W. Lucia, "Undetectable finite-time covert attack on constrained cyber-physical systems," *IEEE Transactions on Control of Network Systems*, vol. 9, no. 2, pp. 1040–1048, 2022.

[37] K. Daria and J. Ahmad, "Predicting and detecting cyber-attacks on industrial control systems using machine learning," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 5, pp. 2959–2967, 2019.

[38] H. H. Alhelou, M. Hamedani-Golshan, T. T. Njenda, and P. Siano, "Decentralized control strategy for detecting and mitigating cyber-attacks on smart grids," *IEEE Access*, vol. 7, pp. 152025–152039, 2019.

[39] L. Wang and M. Liu, "Support vector machine-based detection method for cyber-physical systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 96–107, 2021.

[40] F. Li and Y. Zhang, "Reliable attack detection in dc microgrids using adaptive sliding mode observers," *IEEE Transactions on Smart Grid*, vol. 14, no. 3, pp. 2000–2012, 2023.

[41] M. Adeli and A. Ramezani, "Optimized adaptive observers for cyber-physical attack detection," *IEEE Transactions on Industrial Electronics*, vol. 69, no. 6, pp. 5948–5958, 2022.

[42] H. Zhang and Y. Chen, "Distributed sliding mode observer-based cyber-attack detection in power systems," *IEEE Transactions on Power Systems*, vol. 36, no. 2, pp. 1506–1516, 2021.

[43] M. Brown and Y. Xie, "Sliding mode observer-based detection in industrial control systems," in *Proceedings of the IEEE Conference on Decision and Control (CDC)*, pp. 6023–6028, 2013.

[44] L. Zhang and Z. Wang, "Moving target defense for cyber-physical systems: Coding schemes and applications," *IEEE Transactions on Information Forensics and Security*, vol. 13, no. 5, pp. 1265–1278, 2018.

[45] W. Huang and F. Guo, "Adaptive coding techniques in moving target defense for cyber-physical systems," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 6, pp. 4481–4491, 2020.

[46] M. Yoon and T. Cho, "Coded moving target defense against replay attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 12, no. 2, pp. 1964–1975, 2021.

[47] Y. Mo and B. Sinopoli, "Secure control against replay attacks using watermarking," *IEEE Transactions on Automatic Control*, vol. 54, no. 5, pp. 1093–1099, 2009.

[48] M. Peiravi and A. Nassiri, "Secure signal watermarking for cyber-physical systems," *IEEE Access*, vol. 9, pp. 106512–106523, 2021.

[49] R. Alavi-Khan and H. Shamsi, "Signal watermarking techniques for enhancing cps security against undetectable attacks," *IEEE Transactions on Industrial Informatics*, vol. 18, no. 3, pp. 1729–1739, 2022.

[50] X. Wang and J. Jiang, "Challenges in model-based attack detection for cyber-physical systems," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 50, no. 5, pp. 1085–1098, 2020.

[51] Y. Yilmaz and S. Uludag, "Mitigating iot-based cyberattacks on the smart grid," in *2017 16th IEEE International Conference on Machine Learning and Applications (ICMLA)*, pp. 517–522, IEEE, 2017.

[52] M. Caselli, E. Zambon, J. Amann, R. Sommer, and F. Kargl, "Specification mining for intrusion detection in networked control systems," in *USENIX Security Symposium*, pp. 791–806, 2016.

[53] N. Bakalos, A. Voulodimos, N. Doulamis, A. Doulamis, A. Ostfeld, E. Salomons, J. Caubet, V. Jimenez, and P. Li, "Protecting water infrastructure from cyber and physical threats: Using multimodal data fusion and adaptive deep learning to monitor critical systems," *IEEE Signal Processing Magazine*, vol. 36, no. 2, pp. 36–48, 2019.

[54] A. S. Musleh, G. Chen, and Z. Y. Dong, "A survey on the detection algorithms for false data injection attacks in smart grids," *IEEE Transactions on Smart Grid*, vol. 11, no. 3, pp. 2218–2234, 2020.

[55] R. Qi, C. Rasband, J. Zheng, and R. Longoria, "Detecting cyber attacks in smart grids using semi-supervised anomaly detection and deep representation learning," *Information*, vol. 12, no. 8, p. 328, 2021.

[56] A. Parizad and C. Hatziadoniu, "A laboratory set-up for cyber attacks simulation using protocol analyzer and rtu hardware applying semi-supervised detection algorithm," in *2021 IEEE Texas Power and Energy Conference (TPEC)*, pp. 1–6, IEEE, 2021.

[57] P. Liu, L. Wang, R. Ranjan, G. He, and L. Zhao, "A survey on active deep learning: from model driven to data driven," *ACM Computing Surveys (CSUR)*, vol. 54, no. 10s, pp. 1–34, 2022.

[58] T. Sutharssan, S. Stoyanov, C. Bailey, and C. Yin, "Prognostic and health management for engineering systems: a review of the data-driven approach and algorithms," *The Journal of engineering*, vol. 2015, no. 7, pp. 215–222, 2015.

[59] Y.-J. Liu, J. Li, S. Tong, and C. P. Chen, "Neural network control-based adaptive learning design for nonlinear systems with full-state constraints," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 7, pp. 1562–1571, 2016.

[60] D. Vrabie and F. Lewis, "Neural network approach to continuous-time direct adaptive optimal control for partially unknown nonlinear systems," *Neural Networks*, vol. 22, no. 3, pp. 237–246, 2009.

[61] H. Modares and F. L. Lewis, "Linear quadratic tracking control of partially-unknown continuous-time systems using reinforcement learning," *IEEE Transactions on Automatic control*, vol. 59, no. 11, pp. 3051–3056, 2014.

[62] T. Ozcan and A. Basturk, "Human action recognition with deep learning and structural optimization using a hybrid heuristic algorithm," *Cluster Computing*, vol. 23, no. 4, pp. 2847–2860, 2020.

[63] F. Akowuah and F. Kong, "Real-time adaptive sensor attack detection in autonomous cyber-physical systems," in *2021 IEEE 27th real-time and embedded technology and applications symposium (RTAS)*, pp. 237–250, IEEE, 2021.

[64] F. Farivar, M. S. Haghighi, S. Barchinezhad, and A. Jolfaei, "Detection and compensation of covert service-degrading intrusions in cyber physical systems through intelligent adaptive

control," in *2019 IEEE International Conference on Industrial Technology (ICIT)*, pp. 1143–1148, IEEE, 2019.

[65] W. Li, L. Xie, and Z. Wang, "A novel covert agent for stealthy attacks on industrial control systems using least squares support vector regression," *Journal of Electrical and Computer Engineering*, vol. 2018, no. 1, p. 7204939, 2018.

[66] X. Shao, L. Xie, C. Li, and Z. Wang, "A covert attack detection strategy combining physical dynamics and effective features-based stacked transformer for the networked robot systems," *Nonlinear Dynamics*, vol. 112, no. 21, pp. 19201–19221, 2024.

[67] F. Pasqualetti, F. Dörfler, and F. Bullo, "Attack detection and identification in cyber-physical systems," *IEEE transactions on automatic control*, vol. 58, no. 11, pp. 2715–2729, 2013.

[68] A. Baniamerian, K. Khorasani, and N. Meskin, "Determination of security index for linear cyber-physical systems subject to malicious cyber attacks," in *2019 IEEE 58th Conference on Decision and Control (CDC)*, pp. 4507–4513, IEEE, 2019.

[69] R. S. Smith, "A decoupled feedback structure for covertly appropriating networked control systems," *IFAC Proceedings Volumes*, vol. 44, no. 1, pp. 90–95, 2011.

[70] W. Li, L. Xie, and Z. Wang, "Two-loop covert attacks against constant value control of industrial control systems," *IEEE Transactions on Industrial Informatics*, vol. 15, no. 2, pp. 663–676, 2018.

[71] S. Dilek, H. Çakır, and M. Aydın, "Applications of artificial intelligence techniques to combating cyber crimes: A review," *arXiv preprint arXiv:1502.03552*, 2015.

[72] F. Miao, Q. Zhu, M. Pajic, and G. J. Pappas, "Coding schemes for securing cyber-physical systems against stealthy data injection attacks," *IEEE Transactions on Control of Network Systems*, vol. 4, no. 1, pp. 106–117, 2016.

[73] S. Fang, K. H. Johansson, M. Skoglund, H. Sandberg, and H. Ishii, "Two-way coding in control systems under injection attacks: From attack detection to attack correction," in *Proceedings of the 10th ACM/IEEE International Conference on Cyber-Physical Systems*, pp. 141–150, 2019.

[74] A. Shostack, *Threat modeling: Designing for security*. John Wiley & Sons, 2014.

[75] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *2009 47th annual Allerton conference on communication, control, and computing (Allerton)*, pp. 911–918, IEEE, 2009.

[76] R. S. Smith, "Covert misappropriation of networked control systems: Presenting a feedback structure," *IEEE Control Systems Magazine*, vol. 35, no. 1, pp. 82–92, 2015.

[77] S. Bouabdallah, "Design and control of quadrotors with application to autonomous flying," tech. rep., Epfl, 2007.

[78] M. Ranjbaranhesarmaskan, *Fault recovery of an under-actuated quadrotor aerial vehicle*. PhD thesis, Concordia University, 2010.

[79] M. A. Alsharif, Y. E. Arslantas, and M. S. Hölzel, "A comparison between advanced model-free pid and model-based lqi attitude control of a quadcopter using asynchronous android flight data," in *2017 25th Mediterranean Conference on Control and Automation (MED)*, pp. 1023–1028, IEEE, 2017.

[80] A. E. Lim and X. Y. Zhou, "Stochastic optimal lqr control with integral quadratic constraints and indefinite control weights," *IEEE Transactions on Automatic Control*, vol. 44, no. 7, pp. 1359–1369, 1999.

[81] P. C. Young and J. Willems, "An approach to the linear multivariable servomechanism problem," *International journal of control*, vol. 15, no. 5, pp. 961–979, 1972.

[82] A. S. Tummala and R. K. Inapakurthi, "A two-stage kalman filter for cyber-attack detection in automatic generation control system," *Journal of Modern Power Systems and Clean Energy*, vol. 10, no. 1, pp. 50–59, 2021.

[83] G. Welch, G. Bishop, *et al.*, "An introduction to the kalman filter," 1995.

[84] H. F. Albinali and A. Meliopoulos, "Hidden failure detection via dynamic state estimation in substation protection systems," in *2017 Saudi Arabia Smart Grid (SASG)*, pp. 1–6, IEEE, 2017.

[85] S. Hochreiter, Y. Bengio, P. Frasconi, J. Schmidhuber, *et al.*, "Gradient flow in recurrent nets: the difficulty of learning long-term dependencies," 2001.

[86] Z. Shi, A. A. Mamun, C. Kan, W. Tian, and C. Liu, "An lstm-autoencoder based online side channel monitoring approach for cyber-physical attack detection in additive manufacturing," *Journal of Intelligent Manufacturing*, pp. 1–17, 2023.

[87] A. Thangasamy, B. Sundan, and L. Govindaraj, "A novel framework for ddos attacks detection using hybrid lstm techniques.," *Computer Systems Science & Engineering*, vol. 45, no. 3, 2023.

[88] D. Stiawan, A. Bardadi, N. Afifah, L. Melinda, A. Heryanto, T. W. Septian, M. Y. Idris, I. M. I. Subroto, R. Budiarto, *et al.*, "An improved lstm-pca ensemble classifier for sql injection and xss attack detection.," *Computer Systems Science & Engineering*, vol. 46, no. 2, 2023.

[89] H. H. Htun, M. Biehl, and N. Petkov, "Survey of feature selection and extraction techniques for stock market prediction," *Financial Innovation*, vol. 9, no. 1, p. 26, 2023.

[90] L. Hua, C. Zhang, W. Sun, Y. Li, J. Xiong, and M. S. Nazir, "An evolutionary deep learning soft sensor model based on random forest feature selection technique for penicillin fermentation process," *ISA transactions*, vol. 136, pp. 139–151, 2023.

[91] N. Thacker and A. Lacey, "Tutorial: The likelihood interpretation of the kalman filter," *TINA Memos: Advanced Applied Statistics*, vol. 2, no. 1, pp. 1–11, 1996.