Data-Driven Covert Attack Design and Resilient Detection Using Reinforcement Learning in Cyber-Physical Systems

Anas Berbar

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 & Computer Engineering (MASc)) at

Concordia University

Montréal, Québec, Canada

December 2025

CONCORDIA UNIVERSITY

School of Graduate Studies

This is to certify that the thesis prepared

By: **Anas Berbar**

Entitled: **Data-Driven Covert Attack Design and Resilient Detection Using Reinforcement Learning in Cyber-Physical Systems**

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

Master of Applied Science (Electrical & Computer Engineering (MASc))

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.

_____ External Examiner
Dr. Hamid Taghavifar

_____ Examiner
Dr. Rastko Selmic

_____ Supervisor
Dr. Khashayer Khorasani

Approved by _____
Dr. Abdelwahab Hamou-Lhadj, Chair
Department of Electrical and Computer Engineering

_____ 2026

_____
Mourad Debbabi, Dean
Faculty of Engineering and Computer Science

Abstract

Data-Driven Covert Attack Design and Resilient Detection Using Reinforcement Learning in Cyber-Physical Systems

Anas Berbar

Cyber-Physical Systems (CPS) are becoming more susceptible to sophisticated attacks that take advantage of both physical and cyber components to degrade system operation while remaining stealthy. Two significant contributions to the topic of CPS security are presented in this thesis. First, a brand-new Data-Driven Covert-Replay Attack (DDCRA) is introduced, which uses recorded input-output data to control the system while using covert signal cancellation to avoid detection. The DDCRA is extremely useful and practical in real-world situations because it doesn't require knowledge of the system model or estimation technique, in contrast to conventional covert attacks. Second, a dual-agent reinforcement learning detection architecture is presented to counter such attacks. It consists of a Deep Q-Network (DQN) agent at the command and control center and a Deep Deterministic Policy Gradient (DDPG) agent at the plant side. The DQN agent completes the final attack detection after receiving anomaly signals from the DDPG agent, which keeps an eye on system behavior locally. Simulations on a quadruple-tank process are used to verify the effectiveness of the suggested assault and detection system. The results show that while the suggested RL-based detector greatly increases detection accuracy in previous stealthy scenarios, the DDCRA can evade detection from conventional residual-based techniques. This study enhances the state-of-the-art in the area of intelligent CPS security mechanisms and emphasizes the necessity for more hybrid solutions in the future.

Contents

List of Figures

List of Tables

Chapter 1

Introduction and Literature Review

Cyber-physical systems (CPSs) are systems that integrate physical plant with controllers through communication networks. CPSs have become vital systems in a number of industries, from smart grids and industrial automation to autonomous cars and medical systems. These systems operate under real-time constraints, with an actual connection of software and physical components. Although these capabilities are considered high efficiency and intelligent, they also come with difficult security issues. In contrast to traditional IT systems, the security of CPSs when compromised is a question of not only data breach but also public safety and even human life because of its features such as hybrid dynamics, resource limitations, precise timing requirements, and safety-critical operations, which would lead to traditional cybersecurity defense methods frequently failing when applied to CPS. Additionally, attackers are changing their tactics more and more, taking advantage of the interdependence between the physical and cyber levels to conduct high-impact, complex, and covert attacks that can avoid traditional detection techniques [1].

Modern control systems are the foundation of many important applications in industrial automation, robotics, aviation, and CPS. These systems aim to regulate the behavior of dynamic processes by continually measuring outputs and calculating inputs that guide the system toward desired performance targets like stability, tracking, or disturbance rejection [2].

A state-space formulation, which represents the dynamics of the system as a collection of first-order differential or difference equations, is commonly used to model the behavior of modern control systems. A key element in modern control theory, state estimate is essential to CPS decision-making and feedback control. Due to sensor constraints, cost, or accessibility, not all internal states in many real-world systems can be directly measured. As a result, estimators—also called observers—are used to recreate the internal state of the system using the input-output data that is at hand. It is crucial to evaluate the system's observability before creating a state estimator. A key idea in control theory is observability, which establishes whether a dynamical system's internal states can be deduced from its outward outputs over time. State estimators are often categorized according to the existence of noise or uncertainty and the type of system model (linear vs. nonlinear) [3]. The Luenberger observer is one of the most well known methods that works for noise-free LTI systems. Though it is simple to implement, however it is limited to noise-free environments [4]. A dynamic system's internal states can be estimated from a set of noisy observations using the Kalman filter, an optimum recursive method that is usually described in state-space form. It operates in two phases: 1-prediction, which employs internal system dynamics to predict the present state, and 2-update, which utilizes the new measurement to verify the prediction's accuracy. It is widely employed in robotics, signal processing, control systems, and navigation (including GPS and aircraft).Its advantages include effectiveness, real-time functionality, and optimal performance in the presence of Gaussian noise. Nonetheless, in cases that are nonlinear or non-Gaussian noise, its precision becomes inaccurate [5]. The Extended Kalman Filter (EKF), which is a nonlinear version of the standard Kalman Filter, is designed to manage systems that have nonlinear state transition or measurement models. Utilizing a first-order Taylor expansion to linearize the nonlinear functions about the current estimate allows for the approximate use of the Kalman filter equations. In practical applications where system dynamics are nonlinear, such as autonomous cars, robot localization, and sensor fusion, EKF is frequently used. The main advantage is the ability to apply kalman filtering to non-linear problems with lower computational cost. However, it suffers from linearization-related approximation errors, which can cause divergence or poor performance in highly nonlinear or poorly described systems [6].

In addition to reach optimal states and stability, control techniques in CPS must be resilient to

disruptions, model uncertainty, and possible cyberthreats. It is essential to confirm that the system is controllable before creating a control setup. The ability to drive the system's state to any desired value within a finite amount of time using appropriate control inputs is known as controllability. Controllability ensures that the control designer has total control over the dynamics of the system [3]. In all control systems, before using any control methods to drive the states to a desired value like the Linear Quadratic Regulator (LQR) or its extensions (such as LQI explained in the next section), controllability must be confirmed. This is because controllability defines whether its possible to drive all the states to a desired value. For LTI systems, the LQR finds the control input value which minimizes a cost function, balancing control with optimizing system performance. In order to calculate the state-feedback gain matrix that pushes the system toward stability while reducing deviation from intended behavior, it solves the algebraic Riccati equation. Because LQR can assure stability and excellent performance under well-defined parameters, it is frequently used in industrial automation, robotics, and aircraft. Its main benefits is the stable performance for LTI systems. Its limitations, however, are that it can solely be applied to linear systems until it is enhanced using methods like gain scheduling or nonlinear adaptations, and it necessitates complete state feedback and accurate modeling [7]. The Linear Quadratic Integral (LQI) controller is an extension of the LQR controller in the presence of disturbances or setpoint-reference changes. It adds integral action into the cost function to remove steady-state errors . This is achieved by incorporating an additional state that represents the integral of the output error, so enabling the controller to more precisely follow reference signals. LQI improves the tracking ability of LQR while also retains the optimality and stability guarantees of LQR. It is useful in applications in which precise referencing is required, like servo systems, process control, and tracking problems. The main advantage of LQI lies in its ability to achieve zero steady-state error for step inputs. However, it increases system order due to the added integrator [8].

CPS signifies the integration of physical processes with computational and communication functionalities. They serve as the foundation for many vital infrastructures, such as driverless cars, smart power grids, industrial control systems, and medical equipment. CPS creates a feedback loop between the cyber and physical realms by using embedded controls and sensors to gather data, make choices, and initiate responses in real time.

In contrast to traditional information systems, CPS are closely linked to physical dynamics that continuously transmit and receive data to the command and control center via a network. They are therefore vulnerable to certain types of attacks [9], [10]. Figure 1.1 shows the base architecture of a CPS that is vulnerable to cyber-attacks in the communication channels (Network), while exchanging data from the plant to the command and control center.

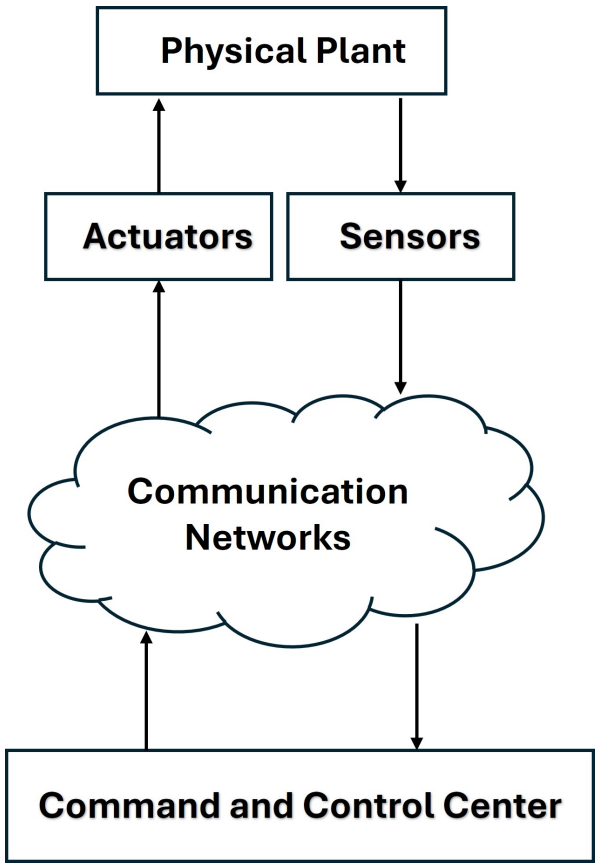

Figure 1.1: General CPS Architecture

Importantly, the design and effectiveness of cyber attacks against CPS are highly dependent on the attacker's knowledge of the system. Cyber-physical security evaluated through these three foundational principles:

- **Confidentiality:** Guarantees that only individuals with permission can access information. if confidentiality is breached, The data would be available to the attackers.

- **Integrity:** Refers to the guarantee that no unauthorized changes have been made to data. Since altered control signals or fabricated sensor data can lead to improper physical actions and potentially dangerous outcomes, integrity is crucial in CPS.
- **Availability:** Guarantees that information and system features are accessible to authorized users when needed.

For accurate cyber attacks modeling, the types of information available to an attacker can be classified as follows [11]:

- **Disclosure Resources:** These point to the attacker's capacity to observe signals from the system. This specifically refers to having read access to the inputs and/or outputs of the system. Disclosure resources allow the attacker to observe system behavior or estimate internal states for more intelligent attack techniques, hence compromising confidentiality.
- **Disruptive Resources:** These resources jeopardizes integrity by giving the attacker the ability to disrupt the signals of the system.
- **Plant Model Knowledge:** This includes the attacker's knowledge of the internal dynamics of the system, which is usually characterized by access to the system matrices A, B, C, and D. With this information, the attacker can create complex model-based attack plans and avoid model-based detection methods.

Modeling threat scenarios, assessing system vulnerabilities, and creating efficient security and detection systems in CPS all depend on an understanding of the categorization of attacker resources. The main categories of attacks are covered in depth after a thorough literature study of the state-of-the-art CPS detection technologies.

1.1 Detection Methods in Cyber-Physical Systems

CPS security is dependent not only on securing communication channels and access control, but also on the ability to detect anomalies and cyberattacks in real time in the case of unfortunate

attacks. Model-based, data-driven, and hybrid detection methods are the three major categories. This section describes the most common detection mechanisms in CPS security.

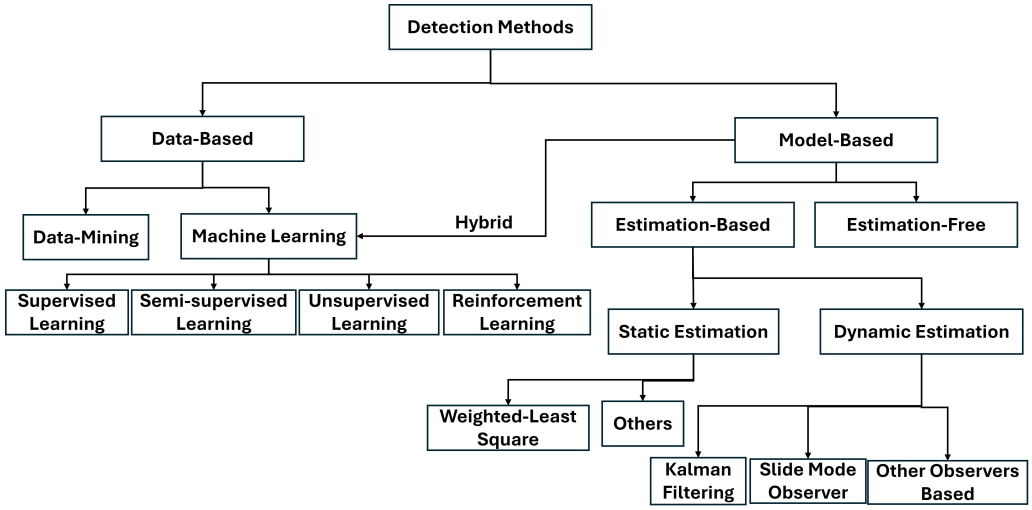

Figure 1.2: Detection Methods Classification

1.1.1 Model-Based Detection Methods

Model-based approaches make use of the dynamics of the system to identify differences between observed and expected behavior. In model-based methods, there are two types of methods: state-estimator-based and estimation-free. The estimation-free methods focus on input-output consistency, system limitations, or statistical consistency. For example, parity-space methods use algebraic redundancy in the system equations to identify inconsistencies, whereas set-based approaches specify acceptable operating constraints and mark any breach as unusual. Techniques for frequency-domain or spectrum analysis use variations in the dynamic response characteristics of the system to find anomalies. These strategies are especially effective in situations where state estimation is infeasible due to low observability, model uncertainty, or computing restrictions [12].

One of the most popular approaches for detecting anomalies or cyberattacks in cyber-physical systems is model-based detection using state observers. The term ”static estimation” describes techniques that use only current or snapshot data to estimate system variables (such outputs or parameters) at a particular moment in time, ignoring how the system changes over time. These

approaches presume a memory-less link between inputs and outputs. For time varying systems, static estimators are usually faster and simpler, but they are less accurate.

On the other hand, dynamic estimating uses knowledge about the dynamics of the system such as differential equations, to track system states or variables across time. Dynamic estimators, like Kalman filters, observers, or particle filters, update the state estimate recursively using input and output data from the past and present.

An observer can lead to generate a residual signal, such as a Kalman filter, that employs a mathematical model of the system to estimate its internal state by comparing the actual measured output to the estimated output produced by the observer. The system would be functioning normally when these residuals are modest and statistically consistent. Significant variances, however, point to the existence of abnormalities that could be the result of errors or cyberattacks. Statistical detection approaches are based on this residual-based method.

The chi-square test offers acceptable limits for statistical consistency and is commonly used to evaluate these residuals.To balance the sensitivity to attacks with the risk of triggering unnecessary alarms, a detection threshold is set according to a chosen false alarm rate.A higher threshold reduces false alarms but risks missing minor inconsistencies that could result in undetectable attacks in this area, whereas a lower threshold improves attack detection but raises the likelihood of false positives. Defenders can adjust detection performance with risk tolerance and anticipated threat levels. All things considered, this technique offers a structured and probabilistic framework to detect attacks [13]. In order to identify and prevent cyberattacks, especially deception attacks, observer-based sliding mode detection in CPS security combines state observation with the resilience of sliding mode control. While a sliding surface is intended to draw attention to differences between expected and actual system behavior, a state observer evaluates the underlying system states in real time. When the system deviates from the sliding surface, it hints that sensor or actuator signals being compromised, which initiates detection. This approach is ideal for CPS contexts where robustness and quick reaction are crucial because it is extremely resistant to system uncertainties and disruptions, such as in [14] and [15]. Unknown Input Observers (UIO) are resilient to unknown disturbances and are intended to isolate defects or attacks. By separating inputs, UIOs increase vulnerability to cyberattacks. By separating the impact of unknown inputs from the residual—that is,

the discrepancy between measured and estimated outputs—they are used to identify abnormalities or malicious signals [16].

1.1.2 Data-Driven Methods

Data-driven techniques include machine learning and data mining. Data mining in CPS detection involves using computational methods to identify patterns and anomalies in extensive plant datasets. This enables the detection of cyberattacks without needing formal models of physical systems. To detect variations from anticipated behavior, methods such as clustering, classification, and anomaly detection are commonly applied to sensor outputs, control inputs, and communication data. This approach is effective because it takes into account both historical and present data. It relies on high-quality labeled datasets,however, it maybe susceptible to problems such as model degradation in rapidly evolving situations and false positives [17]. In [18], to predict students final exam scores and detect high-risk individuals who may drop out, they explored data mining techniques. The research found that incorporating previous academic performance and student engagement metrics into artificial neural networks significantly improved prediction precision, while demographic data by itself was inadequate for succesful forecasting. The study in [19] identified important characteristics pertinent to heart disease prediction using data mining techniques. Seven classification algorithms—logistic regression, naive Bayes, support vector machines, a hybrid voting approach that combines naive Bayes and logistic regression, neural networks, decision trees, and k-nearest neighbors—were used to build predictive models. Furthermore, the combination of synthetic minority oversampling techniques and data mining techniques, as mentioned in [20], mproved the precision of heart failure patients' survival forecasts.

Data-driven machine learning (ML) techniques use learning patterns found in system-generated data to train machines to carry out challenging tasks. The quantity and quality of input data have a major impact on how well these algorithms work. ML techniques are typically divided into three categories based on how the machine learns from this data: supervised learning, unsupervised learning, and reinforcement learning. Below is a quick summary of various techniques.

Supervised learning is the most common approach when labeled datasets of normal and attack behaviors are available. Algorithms like Linear Regression (LR) as in [21], Random Forests

(RF) as in [22], Extended Nearest Neighbor(ENN) as in [23] overcoming some challenges such as prelabeled sample density and distribution in k-nearest neighbor(KNN) [24], the Extreme Learning Machine(ELM) [25] to overcome the problems of extensive training time required in Auto Encoders (AE) [26], Support Vector Machines (SVM) in [27] which has huge training time, Decision Trees (DR) in [28] which is easy to build but has an overfitting problem, Artificial Neural Networks (ANNs) are modeled after the functioning of the human brain [29] and are widely used for tasks such as function approximation, estimation, and classification [30]. ANNs can be structured with either a single or multiple hidden layers [31] and [32], and their output is determined by a combination of activation functions, biases, and the weighted sum of inputs to each neuron [33].

Convolutional neural networks (CNNs) use convolution operations inside their layers, in contrast to conventional neural networks that rely on matrix multiplication. CNNs are effective in image processing and pattern recognition applications because of this architecture, resulting in efficient extraction of different features from input data [34].

Deep Neural Networks (DNNs), a kind of neural network made up of several hidden layers, have been successfully used in a variety of challenging learning tasks Because of their high accuracy [35].

By utilizing both a big amount of unlabeled data and a limited collection of labeled data, semi-supervised learning fills the gap between supervised and unsupervised learning. Self-training and graph-based approaches are two strategies that can enhance detection in actual CPS situations [36], [37] and [38].

When attack labels are not provided, unsupervised learning is essential for anomaly identification in CPS. To simulate typical system behavior, clustering methods (such as K-Means or DBSCAN), dimensionality reduction (such as PCA), and autoencoders are used. Any notable departure from this model is identified as a possible abnormality. A particular time series model called the Hidden Markov Model (HMM) is frequently used in FDI detection to predict sample trends [39]. A kind of feedforward neural networks (FNN), the Probabilistic Neural Network (PNN) is mostly used for classification and pattern recognition applications and provides quicker processing speeds than multilayer perceptron networks [40]. By starting Deep Belief Networks (DBN) with already learned weights, training time can be reduced. Backpropagation can then be used to fine-tune the network into a generatively pre-trained deep neural network (DNN) [41]. An efficient method for

detecting outliers is the Isolation Forest (IF), which is made up of several isolation trees. High-dimensional data increases its performance [42]. All things considered, even though these methods might produce similar results, each has unique traits and working mechanisms.

Reinforcement learning (RL) uses interactions with the CPS environment to teach agents the best defense strategies. Proactive defense techniques including dynamic reconfiguration, access restriction, and real-time attack response can be implemented with RL. CPS can gradually increase their robustness on their own thanks to algorithms like Q-learning and Deep Q-Networks (DQN), but issues like real-time restrictions need to be properly handled. RL is going to be used in this thesis, hence a more focused literature review will be addressed in the third chapter.

1.1.3 Hybrid Detection Frameworks

Hybrid schemes combines these approaches of ML with model-based. Hybrid detection schemes can enhance detection accuracy and robustness. To identify FDI attacks in power systems CPS, a study by [43] provided a hybrid detection technique that combines an enhanced Adaptive Kalman Filter (AKF) with a GRU-CNN neural network. While the GRU-CNN extracts temporal and spatial information from the data to increase detection performance, the AKF component offers model-based residual analysis. A Hybrid Metaheuristics-based dimensionality reduction with Deep Learning for FDI detection (HMDR-DLFDIA) was presented in [44], which integrated optimization methods for feature selection with a stacked autoencoder for anomaly detection. This approach demonstrates high accuracy in identifying FDIAs in smart grid systems.

1.1.4 Types of Cyber Attacks on CPS

False Data Injection (FDI) Attacks

One of the most researched and useful types of attacks is FDI. To trick the state estimator or controller, they are introducing deliberately constructed harmful data into sensor readings or actuator orders. Distorting the system's observed status while evading detection is the main goal. Disruptive resources at the input or output channel should be present in a simple detectable FDI attack. However, given the plant's model knowledge, a more sophisticated form of FDI attacks

can be carried out. A structured approach for creating FDI attacks that can evade Bad Data Detection in state estimation procedures was provided in [45]. The measurement matrix and topology of the system are considered to be accessible to the attacker, enabling them to create undetectable attack paths. The study presents this attack design as a restricted optimization problem and uses simulations to show that successful, covert interruptions can result from even a small amount of attacker knowledge. The findings draw attention to CPS weaknesses and the inadequacy of conventional detection techniques in the face of structured FDI attacks. A study of FDI attacks that target the input channel—more precisely, the controller-to-actuator link—in linear time-invariant cyber-physical systems governed by a Linear Quadratic Gaussian (LQG) controller was presented in [46]. The attacker's primary objective is to undetectably reduce system performance as determined by the Kalman filter's state estimate inaccuracy. The authors define an epsilon-stealthy attack using the Kullback-Leibler divergence, guaranteeing that the statistical difference between normal and attacked innovation sequences stays below a threshold epsilon, in order to expand the concept of stealthiness beyond particular detectors like the chi-square test. The attacker is forced to operate under a constraint as a result. In order to optimize the long-term performance loss while adhering to the epsilon-stealthiness criterion, the authors construct the attack as an optimization problem. They provide two tactics for closed-form attacks. The first approach determines whether the stealthiness criteria is met after optimizing the estimation error. The second approach assesses the ensuing degradation after first guaranteeing the epsilon-stealthiness condition. It is demonstrated that both approaches are optimal under certain system structure-related criteria, especially when a term involving the attack direction and the Kalman gain satisfies certain constraints. In [47], they presented a novel class of stealthy FDI attacks in CPS. This innovative method enables the attacker to copy the target system and generate the necessary signals, in contrast to typical FDI attacks that require observability of the system. In order to make the attack invisible by traditional anomaly detectors such as Kalman filter, this replica runs in parallel and produces attack inputs and outputs. As long as the system is linear and the attacker is fully aware of it, this method's strength is its capacity to precisely maintain the expected residual behavior, successfully tricking detection methods of any complexity. Although the attack is theoretically strong, its viability depends on a number of bold

assumptions, including the attackers complete model knowledge and complete access to the actuator and sensor channels. The study emphasizes the shortcomings of the existing CPS detection methods and the urgent need for defense measures.

Replay Attacks

Replay attacks are a basic and sneaky type of cyberattack on CPS during steady-state operations. In order to trick the system, an attacker records valid sensor outputs during regular steady-state operation and replays them later. These attacks take advantage of the presumption that repeated sensor values indicate accurate command and control value ranges.

One of the main characteristics of replay attacks is their ability to be executed without any knowledge of the estimation algorithm or the system model, including system matrices (A, B, and C). Replay attacks require complete disruptive resources to inject previously recorded data back into the system at the output and control the plant through injected control actions, as well as disclosure resources to the sensor communication channel to monitor and record real-time data. For the duration of the attack, it is assumed that the system is functioning in a steady state.

The replay attacks was introduced in the literature by [48], who researched the circumstances under which such attacks can be stealthy in CPS. Later studies expanded on this literture such as [12]. They studied the effects of replay attacks on system observability and controllability and classified them within a larger class of stealthy attacks. According to their research, replay attacks take advantage of a basic weakness in the system: if watermarks or time-varying authentication are not employed, the system is unable to discern between recorded and live data. By showing that replay attacks can evade anomaly detectors in linear time-invariant systems when the attacker precisely timed the injection of recorded signals, [49] further highlighted the practical risk of replay attacks. In practical applications, replay attacks are a key component of the Stuxnet attack, which concealed the sabotage by manipulating actuator commands in programmable logic controllers (PLCs) while replaying normal sensor readings to monitoring systems [50] and [51].

In order to defend against replay attacks, [52] suggested using physical watermarking, which involves adding tiny, random signals to the control input in order to disturb the system in a known manner. These disruptions appear in the output during regular operation, but during a replay attack,

the attacker's replayed sequence separates the output from the current input. The authors demonstrated how one can statistically determine whether an attack is present by comparing input-output pairs to determine whether the control-induced excitement is absent.

Building on this, [53] presented a generalized framework for dynamic watermarking that balances control performance and detection accuracy by optimally designing excitation signals. Their approach enhanced the watermark energy while adhering to false alarm. This study presented dynamic watermarking as a feasible and demonstrably effective method for detecting replay attacks in LTI systems.

Information-theoretic methods have also been investigated concurrently. For instance, [54] suggested an entropy-based detection technique that depends on keeping an eye on the sensor outputs' Shannon entropy. The lack of fresh system stimulation during a replay attack lowers the measured outputs' entropy, which serves as a foundation for detection. This approach works efficiently with systems having complicated stochastic behavior.

The idea of stealthy attacks was defined inside a system-theoretic framework by [12], which also described the circumstances in which replay and other deception attacks are undetectable. Despite not being exclusively concerned with replay attacks, their work offered basic notions of detectability and identifiability that impacted subsequent detector design. Specifically, they demonstrated that unless active detection is employed such as watermarking, any attack aligned with the system's unobservable subspace can avoid detection.

Model-based methods depend on unpractical model with respect to real-world challenges, which can lead to false alarms[12]. They are only robust against environments assuming gaussian noise. Moreover, replay attacks inherently exploit the model's predictions; if the captured data aligns with the anticipated behavior, as shown by [48], the residual remains minimal and the attack stays stealthy. These techniques also face challenges in scaling because simulating extensive systems leads to synchronization problems and computational costs that complicate real-time operations. Because constant re-identification is expensive and challenging to accomplish, static models are especially unsuitable for time-varying systems. Furthermore, model-based detector deployment frequently necessitates specialized knowledge for calibration and tweaking, resulting in operational overhead. The vulnerability of defenses based solely on models is emphasized by the reality that

attackers who possess access to or understanding of the system model can carry out undetectable attacks by conforming their strategies to the model's assumptions [52]. Due to these constraints, there is growing interest in hybrid or data-driven detection techniques that are better able to adjust to real-world system uncertainties.

Data-driven detection architectures are developed to overcome some of the drawbacks of model-based approaches. In [55], they proposed a framework to identify deviations in system outputs employing temporal difference. The framework compares system history to expected behaviors learnt from training data instead of depending on residuals. The authors show that by utilizing models to check trained patterns, their method not only detects replay attacks with high precision but also prevents false positives.

The application of moving target defense, as presented by [56], is another novel approach. This strategy entails gradually altering a parallel auxillary system controller dynamics or auxillary system parameters at random intervals that are only known to the defender. Because they are inconsistent with the modified system, replay attacks—which rely on previously recorded data corresponding to previous system conditions—are thus revealed. With the new configuration [56], this method turns the detection problem into a time consistency problem, in which the replayed data can be statistically distinguished from real data.

Zero Dynamics Attacks

A family of stealthy cyber-physical attacks known as zero dynamics attacks (ZDAs) takes advantage of control systems' internal dynamics by focusing on their zero dynamics, or modes that have no effect on the output. Attackers can cause huge internal state deviations without changing the system's output by introducing attack inputs that are consistent with the unobservable modes. This allows them to pass the traditional output-based detection methods.

ZDAs were first introduced by [57], who showed that an attacker with complete knowledge of the system model might provide input signals that excite the system's zero dynamics, causing undetectable changes in the internal states of the system. The possible seriousness of ZDAs in networked control systems was brought to light by this seminal work.

In continuous, [12] discussed a detection and identification framework CPS, highlighting the

difficulties presented by attacks that take advantage of unobservable subspaces of the systems such as zero dynamics.

The conditions in which ZDAs can be carried out in [11] has been explained, examining the relationship between system behavior and the possibility of such attacks. They shed light on how sensor placement and system architecture can affect the system's vulnerability.

Researchers have studied the threat posed by ZDAs in the context of nonlinear systems and have developed generalized attack strategies that work against nonlinear multiple-input multiple-output CPS. These experiments show that attackers can create stealthy ZDAs even in nonlinear contexts, making detection more challenging.

Numerous detection and mitigation techniques have been put forth to combat ZDAs. Auxiliary systems and event-triggered communication schemes are two methods. In order to identify and isolate ZDAs, [58] created a detection method that uses an auxiliary system with self-triggered communication, even in situations when attackers are fully aware of the dynamics of the auxiliary system and are able to initiate FDI attacks on all communication channels.

An additional strategy is to change the system's sampling techniques. In the literature, employing generalized hold functions instead of zero-order holds can mitigate the effects of sampling-induced ZDAs by assigning sampling zeros to stable areas, as shown in [59].

Additionally, data-driven strategies have been investigated. In order to detect both traditional and enhanced ZDAs by tracking changes in state values, [60] suggested an autoregressive model that creates data linkages between original and fake data.

In [61], they investigated periodic ZDAs in multi-agent systems without velocity measurements and found that such attacks can interfere with agent consensus. They emphasized the significance of system attributes and network architecture in attack resilience by offering adequate settings to measure the effect of ZDAs on system consensus.

Covert Attacks

The covert attack represents a significant threat due to high stealthiness against detectors that remain at the command and control side. Because it can manipulate the system by introducing data at the input while simultaneously eliminating its effect at the output, this attack is especially risky.

Disruptive resources at the input and output are necessary for the covert attack [62]. Conventional covert attack models usually make the assumption that the attacker has complete control over all sensor outputs and control inputs. However, in reality, attackers frequently have limited resources [11]. An attack is referred to as a limited covert attack (LCA) when only a portion of the control and/or sensor channels can be penetrated. As previously shown by [63], when the attacker uses a decoupling approach to cancel their effect on limited outputs, LCAs can maintain their stealth.

By adding an extra dynamic system to the original system, [64] suggested a unique detection model-based approach. By altering the system's internal structure, this augmentation makes sure that even covert attacks—such as those that coincide with zero dynamics—produce detectable residuals from the auxiliary system that the attacker is unlikely to be able to identify. Their approach is based on carefully crafting the auxiliary system so that any coordinated manipulation of inputs and outputs by an attacker would be inconsistent with the dynamics of the auxiliary system and cause anomaly warnings.

UIOs which are intended to estimate system states while being detached from unknown or adversarial inputs, are another successful model-based technique. By projecting system outputs into a subspace orthogonal to the attack vector, [11] used this technique to isolate malicious interventions. The UIO ensures detection even when there is model uncertainties by observing the differences between expected and observed behavior when the attack attempts to stay stealthy. Moreover, their secure control framework demonstrated that detectability is directly effected by system architecture regards to the number and placement of sensors.

A distributed model-based detection approach appropriate for large-scale interconnected systems was presented in more recent work by [65]. To keep an eye on system consistency, their approach makes use of inter-subsystem communication and local models. In order to detect covert attacks without the need for global observability or centralized monitoring, each subsystem independently evaluates its status and compares its anticipated outputs with nearby subsystems. Large, modular CPS, like power grids or transportation systems, benefit greatly from this distributed architecture. However, it mostly depends on trustworthy inter-subsystem communication and precise local models.

A detection framework for LCAs was presented in [63]. Their strategy is based on creating

a virtual decoupling mechanism that forces the attacker to stop from aligning attack signals with the system's zero dynamics. Their approach improves detection capability even when the attack vector is limited by rearranging the observer dynamics and incorporating decoupling matrices. The decoupling method makes sure that the zero dynamics of the system cannot hide the attack-induced inconsistencies.

Denial-of-Service Attacks

In CPS, a Denial of Service (DoS) attack is simply an interruption of communication. The DoS attacks are extremely dangerous, For instance, [11] examined denial-of-service attacks that prevent sensor measurements, demonstrating how persistent denial can cause linear systems to become unstable by depriving the controller of feedback. Similarly, [66] shown that by disrupting sensor updates, timed DoS attacks on water treatment systems might result in tank overflows or chemical imbalances. [67] suggested employing statistical anomaly detection on data packet inter-arrival timings to detect such attacks, successfully identifying lengthy silences typical of DoS. In a different approach, [12] employed system-theoretic techniques to monitor variations in system observability and manage energy requirements, allowing the system to issue alerts during data loss caused by an attack.Table 1.1 summarizes the different papers discussed with its attack type, detection type, and their focus. Meanwhile, Table 1.2 summarizes the attacks with knowledge required to initiate.

Table 1.1: Summary of Cyber-Physical Attacks and Detection Methods

Reference	**Attack Type**	**Detection Type**	**Focus**
[48]	Replay	None (modeling)	Formalized replay attacks and conditions for stealthiness
[52]	Replay	Model-Based	Detects decoupling between input-output under replay attack
[53]	Replay	Model-Based	Optimized input excitation for statistical detection (Watermarking)
[54]	Replay	Data-Driven	Lower output entropy under replay signals detection
[57]	ZDA	None (modeling)	Introduced ZDA using unobservable internal dynamics
[64]	ZDA, Covert	Model-Based	Detection of attacks via residuals through an auxillary system
[11]	FDI, ZDA, DoS	Model-Based	Different attacks and residual detection
[63]	Local Covert	Model-Based	Breaks alignment with zero dynamics through Decoupling Observer
[65]	Covert	Model-Based	Cross-checks subsystems to expose inconsistencies by Distributed Estimation
[66]	DoS	Model-Based	Showed physical consequences of timed DoS
[12]	FDI, ZDA, DoS, Replay	Model-Based	Defines residual detection of stealthy attacks
[59]	ZDA	Model-Based	Shifts sampling zeros to stable locations using generalized hold functions
[60]	ZDA	Hybrid Approach	Flags signal inconsistencies via autoregression model
[61]	ZDA	Model-Based	Reveals attack via broken agents agreement between multi-agents
[55]	Replay	Hybrid Approach	Integrating data-driven runtime monitoring with model-based formal verifications

Table 1.2: Comparison of Common Cyber-Physical Attacks

Attack Type	**Model Knowledge Required**	**Disruption Resources Required**	**Disclosure Resources Required**
Covert Attack	Partial to full knowledge of the model and controller	Yes to Input and Output Channels	Linear Systems(No) Non-Linear Systems(Yes to either input or output channels)
Zero Dynamics Attack	Full knowledge of system matrices and invariant zeros	Yes to input channels	No
Replay Attack	No	Yes to input and output channels	Yes to output channels
FDI Attack	No	Yes to either input or output	No

Evaluation of Detection Methods in Cybersecurity for CPS

To evaluate cyberattack detection, the efficiency of cybersecurity detection techniques must be thoroughly assessed before being employed. Standard performance indicators developed from a confusion matrix, which divides detection results into four categories—true positives, false positives, false negatives, and true negatives—are frequently used for this assessment.

Confusion Matrix Definitions

- **True Positive (TP):** An actual cyberattack that is correctly identified by the detection system.
- **False Positive (FP):** A normal behavior incorrectly flagged as an attack.
- **False Negative (FN):** An actual attack that is missed by the detection system.
- **True Negative (TN):** A normal behavior correctly identified as normal.

These outcomes are typically organized in a confusion matrix:

Table 1.3: Comparison of TP, FN, FP and TN

	Predicted: Attack	**Predicted: Normal**
Actual: Attack	True Positive (TP)	False Negative (FN)
Actual: Normal	False Positive (FP)	True Negative (TN)

Performance Metrics From the confusion matrix, the following key metrics from [68] are derived:

Accuracy The ratio of successfully classified cases (true positives and true negatives) to the total number of predictions

Precision Calculates the percentage of accurately detected attacks out of all anticipated attacks:

Recall (Sensitivity) Calculates the percentage of real attacks that are accurately identified:

F1 Score The harmonic mean of precision and recall, used as a balanced performance metric

The range of the F1 score is 0 to 1. Better overall detection performance is indicated by a score closer to 1, especially in settings with unbalanced data (i.e., attacks are uncommon).

Evaluation Procedure To assess a detection method's performance in CPS, the following steps are generally followed:

(1) Simulate or gather information from a CPS in both under attack and regular operating scenarios.

(2) Indicate whether attacks are present or not by labeling the data as true labels (ground truth).

(3) Analyze the data and provide anticipated labels using the wanted to test detection method.

(4) Build the confusion matrix by comparing predicted labels to true labels.

(5) Compute precision, recall, and F1 score using the equations provided in [68].

(6) Compare the results by different detection methods.

Operational Considerations In addition to statistical accuracy, practical concerns must be addressed:

- A high false positive rate may lead to unnecessary defensive actions.
- A high false negative rate means that actual attacks go undetected, jeopardizing system safety.

Therefore, finding the optimal trade-off between precision and recall is essential for securing cyber-physical systems effectively.

By examining CPS vulnerabilities and revealing new threat vectors, this thesis tackles the need for stronger security frameworks. In particular, we present and examine a unique cyberattack that targets the communication data between the CPS's plant and control. By carefully taking advantage of the underlying physics and model-based assumptions, this attack works under the radar of current anomaly detection systems.

This thesis shows how this new form of attack can cause substantial physical disruption while remaining undetected through mathematical modeling and system simulations. In addition to adding to the theoretical underpinnings of CPS security, the thesis highlights the necessity of creating defense measures that are durable, adaptable, and model-aware by exposing these hidden vulnerabilities.

1.2 Summary

In this chapter, we presented the idea of CPS and examined the particular security-related difficulties they encounter. Because CPS closely connects the digital and physical worlds, unlike typical IT systems, cyberattacks may have real-world repercussions. Firstly, the foundational concepts of control theory was discussed, such as observability, controllability, and state estimation, all of which are basic elements that builds the CPS systems and for security to be implemented. Next, we examined how various resources, such as access to sensor data or control channels, might be used by attackers to exploit CPS and how the sophistication of their attacks is determined by these resources. Based on this, we looked at a number of attack types, each backed by important research from the literature, such as FDI, replay attacks, zero dynamics attacks, covert attacks, and denial-of-service

(DoS) attacks. We covered three main detection strategies to counter such attacks: data-driven strategies that employ machine learning, model-based strategies that rely on system dynamics, and hybrid frameworks that mix the two. To assess the effectiveness of various detection algorithms, we also proposed performance criteria such as precision, recall, and F1-score. All things considered, this chapter establishes the foundation for the remainder of the thesis. It draws attention to the pressing need for better detection techniques and serves as inspiration for the innovative attack and detection framework that will be created and examined in the upcoming chapters.

Thesis Contributions

Two major contributions are introduced by this thesis, each of which is covered in its own chapter. Together, they seek to reveal a novel cyber-physical system (CPS) vulnerability and suggest a cutting-edge security strategy that can counter such new attacks.

Chapter 2: A New Data-Driven Covert-Replay Attack

The first significant contribution is the creation of a new cyberattack that combines concepts from replay and covert attacks. This attack is special because it accomplishes the covert attack's stealthy qualities without requiring any understanding of the system model. Rather, it conceals hostile activity from the system's monitoring systems by using recorded input-output data, akin to a replay attack. This is important because, in order to eliminate their impact at the output, covert attacks typically require access to system dynamics or model parameters. Our suggested attack completely eliminates that prerequisite, making it far more feasible and more difficult to identify. Additionally, we provide a hybrid-based detection that deceives the attacker. Nevertheless, we demonstrate that this prevention technique is passable if the attacker knows how it operates.

Chapter 3: Reinforcement Learning-Based Double-Agent Detection

The second contribution is the creation of a detection framework based on reinforcement learning to get over the drawbacks of traditional detection techniques which was shown in the first chapter to be unreliable. This method makes use of two cooperating learning agents:

- A plant-side DDPG agent that gains the ability to comprehend system behavior and produce more informative signals. These signals, together with system observations, are used by a DQN agent at the command and control center to identify anomalous activity.

The purpose of this double-agent strategy is to identify attacks, such as the data-driven covert-replay attack described in Chapter 2, that the system has never encountered. It doesn't rely on preset models or presumptions because it learns directly from interactions with the environment, which makes it more adaptable to unidentified dangers. In conclusion, this thesis presents a scalable and intelligent detection method that can react to future, more complex cyber attacks in addition to exposing a new vulnerability in CPS security through a novel attack.

Chapter 2

Design of Data-Driven Covert Attack

Advanced automation and intelligent control are made possible by the growing integration of CPS with communication and computation components. But this interconnectedness also creates new risks, especially when it comes to cyberattacks that target system data. Among these, data-driven attacks are particularly dangerous because of their intelligence and capacity to affect system behavior while remaining covert.

Data-driven attacks typically take advantage of disclosure resource access to deduce or approximate the system's core model. Equipped with this information, attackers can create malicious inputs or measurement changes that imitate regular operation, getting beyond traditional detection methods like Kalman filters and residual-based anomaly detectors. In addition to impairing performance, these attacks have the potential to force the physical system into dangerous operating areas without sounding an alarm.

This chapter examines the growing threat of data-driven cyberattacks in CPS. It introduces a novel data-driven attack that does not estimate the system's internal states but can still be implemented even in the absence of model knowledge, as well as a model-based detection method designed to trick the attacker into identifying the attack. The work in this chapter is structured as follows: a background of replay and covert attacks, literature review of the current state of research on data-driven and model-based cyberattacks, including false data injection, replay, and covert attacks. Problem Formulation: introducing a novel data-driven attack scenario targeting a closed-loop

networked control system for a two-tank process. Finally, a prevention mechanism against this attack. The goal is to ensure that attacks based on inaccurate data failing to achieve their intended effect or become detectable by conventional anomaly detection schemes.

2.1 Background

2.1.1 Replay Attack

The replay attack is executed in two stages:

1. Recording Phase (Data Collection) In this phase, the attacker passively observes the system during a period of normal operation. No interaction or interference occurs during this time. The attacker continuously records a window of legitimate sensor outputs:

$$\mathcal{Y}_{\text{recorded}} = \{y_{k_0}, y_{k_0+1}, \ldots, y_{k_0+T}\}$$

Here, k_0 is the start of the observation window, and T is the total number of recorded time steps. The attacker aims to collect sensor measurements that are representative of a steady state condition.

2. Attack Phase Once the attacker decides to initiate the attack—typically during a period when the system is expected to behave similarly to the recorded window, as the reference sensor values did not change (same steady state condition from the recording phase), they begin injecting the stored measurements in place of the actual sensor data. This misleads the controller into thinking that the system remains in the same (safe) state, even if it's drifting toward instability or being physically manipulated.

The replayed data is injected as $\tilde{y}_{k^*+i} = y_{k_0+i}$ for $i = 0, 1, \ldots, T$ where k^* is the start time of the replay phase. Replay attacks are particularly difficult to detect using standard statistical methods because the injected data conform to the expected measurement distribution, and the residual signal remains within the nominal confidence bounds. The overall attack is shown in Figure 2.1.

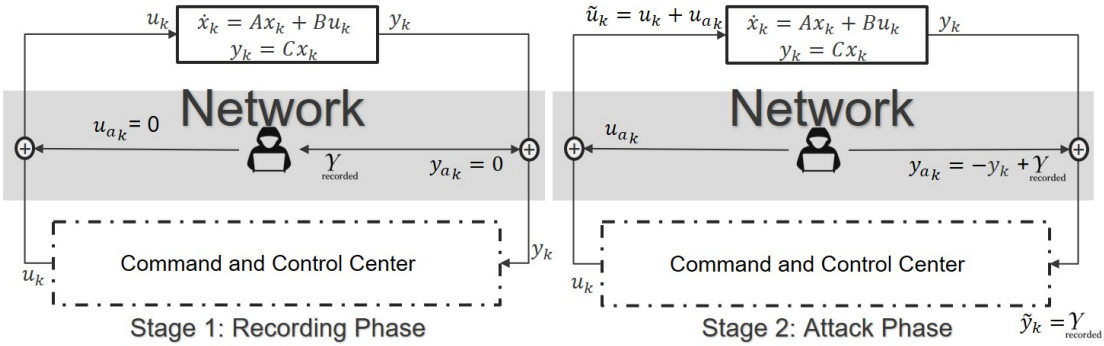

Figure 2.1: Replay Attack

2.1.2 Covert Attack

The original covert attack, as described in [57] relies on full knowledge of the model represented by matrices A, B, C to implement the attack successfully. Its stealthiness is proven as follows: Let $u_a(s)$ and $y_a(s)$ be the Laplace transforms of the attack inputs u_a and y_a respectively. The monitoring system observes the output as

$$\begin{aligned} y'(s)|u_a \neq 0 &= y(s) - y_a(s) \\ &= G(s)(u(s) + u_a(s)) - y_a(s) \end{aligned} \tag{2.1}$$

where $G(s) = C(sI - A)^{-1}B$. By choosing the attack signal $y_a(s) = G(s)u_a(s)$, the output becomes

$$\begin{aligned} y'(s)|u_a \neq 0 &= G(s)(u(s) + u_a(s)) - G(s)u_a(s) \\ &= G(s)(u(s) \end{aligned} \tag{2.2}$$

which proves the attack's complete stealthiness, as the effect of u_a does not appear in the output seen by the monitor.

2.2 Literature Review

The FDI attacks can be designed to be stealthy as in [47]. address the design of false data injection (FDI) attacks that achieve complete stealthiness in cyber–physical systems (CPS). The authors

propose a self-generated approach, allowing attackers to construct FDI signals without relying on real-time system data, thereby enhancing the practicality and stealth of such attacks.

The CPS is modeled as a discrete-time LTI system as in Equations 2.15 and 2.16 without noise or disturbance.

An observer, typically a Kalman filter, estimates the system state based on the output measurements.

The attacker injects false data into the sensor measurements, resulting in:

$$\tilde{y}(k) = y(k) + y_a(k) \tag{2.3}$$

where $y_a(k)$ is the attack signal. The objective is to design $y_a(k)$ such that the attack remains undetected by standard residual-based detectors. The authors define complete stealthiness as the condition where the injected attack does not alter the statistical properties of the residuals used for detection. Mathematically, this implies:

$$r(k) = \tilde{y}(k) - C\hat{x}(k) = y(k) + y_a(k) - C\hat{x}(k) \tag{2.4}$$

To achieve $r(k)$ indistinguishable from the nominal residual, $y_a(k)$ must satisfy:

$$y_a(k) = Ce(k) \tag{2.5}$$

where $e(k) = x(k) - \hat{x}(k)$ is the estimation error. However, since $x(k)$ and $\hat{x}(k)$ are not accessible to the attacker in real-time, the authors propose a self-generated approach.

The proposed strategy involves the attacker constructing a dynamic system that mimics the behavior of the estimation error:

$$e_a(k+1) = (A - KC)e_a(k) \tag{2.6}$$

$$y_a(k) = Ce_a(k) \tag{2.7}$$

Here, K is the observer gain matrix. By initializing $e_a(0)$ appropriately, the attacker can generate $y_a(k)$ that satisfies the stealthiness condition without real-time system data. To further enhance stealth, the authors introduce the concept of energy stealthiness, ensuring that the energy of the attack signal remains bounded:

$$\sum_{k=0}^{\infty} \|y_a(k)\|^2 < \infty \tag{2.8}$$

This condition prevents the attack from causing noticeable anomalies over time. The authors validate their approach through simulations on the IEEE 6-bus power system. The results demonstrate that the self-generated FDI attacks can effectively disrupt the system while remaining undetected by traditional residual-based detectors.

The current state of the art shows that covert attacks can be done with model knowledge as in the original covert-attack paper [57] which is discussed in the background, or in some advanced versions a partial knowledge can be enough to implement a covert attack as will be discussed explicitly in this section. [69] introduces a novel class of cyberattacks on CPS called *Stealthy Targeted Local Covert Attacks*. Unlike traditional stealthy attacks that often require full model knowledge, the proposed attacks are designed to manipulate only a subset of sensors and/or actuators to drive the system into a desired state, and remain undetected (stealthy) by standard anomaly detection mechanisms. In this setp, the attacker injects attack signal $u_a(k)$ into specific control channels and simultaneously injecting $y_a(k)$. Consequently, the actual input and output observed by the controller become:

$$\tilde{u}(k) = u(k) + u_a(k), \quad \tilde{y}(k) = y(k) + y_a(k) \tag{2.9}$$

The attacker's has two main objectives: first, to steer the system state to a specific desired state $x(k_f) = x_{\text{target}}$ at a designated final time k_f; and second, to ensure that the manipulated outputs remain consistent with the nominal outputs for all uncompromised sensor channels, thereby being stealthy. Formally, the stealthiness condition requires that $\tilde{y}_j(k) = y_j(k)$ for all $j \notin \mathcal{Y}_a$, where $\mathcal{Y}_a \subset \{1, \ldots, p\}$ denotes the set of compromised sensor indices.

To determine whether such an attack is feasible, the authors analyze the system's zero dynamics

and derive the conditions for the existence of stealthy attack trajectories. If the system's internal dynamics can be excited in a direction that remains unobservable through the uncompromised outputs, then a covert attack exists.

The attack design is framed as an optimal control problem, where the attacker seeks to minimize the actuation energy of the injected input sequence while achieving the target state and satisfying the stealthiness constraints. The optimization problem is given by:

$$\min_{u_a(0),\ldots,u_a(k_f-1)} \sum_{k=0}^{k_f-1} \|u_a(k)\|^2 \tag{2.10}$$

subject to the system dynamics:

$$x(k+1) = Ax(k) + B(u(k) + u_a(k)), \quad x(0) = x_0, \quad x(k_f) = x_{\text{target}} \tag{2.11}$$

and the stealthiness constraint imposed on $\tilde{y}(k)$.

The authors provide both optimal and suboptimal algorithms to compute attack inputs that satisfy the above criteria. The effectiveness of the approach is demonstrated through numerical simulation on a remotely-controlled helicopter system, where the attacker successfully drives the system to an undesired state without triggering detection alarms on the uncompromised sensors. In [70], they present a data-driven approach to designing optimal stealthy attacks on CPS using only historical input-output data. Unlike model-based strategies, which require knowledge of the system matrices, this method assumes the attacker has access only to previously recorded data, making it highly relevant for realistic attacker scenarios.

The CPS is modeled similar to the previous paper with $\tilde{u}(k) = u(k) + u_a(k), \quad \tilde{y}(k) = y(k) + y_a(k)$. The goal of the attack is to maximize the estimation error of a Kalman filter observer while remaining stealthy. Since the attacker does not know A, B, or C, it uses subspace identification methods to approximate the system's Markov parameters using historical data.

Let $\tilde{\hat{x}}(k)$ denote the Kalman estimate under attack. The attack seeks to maximize the expected squared estimation error:

$$\max_{u_a(0),\ldots,u_a(T)} \mathbb{E}\left[\|x(k) - \tilde{\hat{x}}(k)\|^2\right] \tag{2.12}$$

subject to a stealthiness constraint measured via the Kullback–Leibler (KL) divergence between the distribution of innovation sequences under normal and attacked conditions. Let $\nu(k)$ be the innovation sequence of the Kalman filter, and let $P_{\tilde{\nu}}$ and P_{ν} be the respective distributions under attack and nominal behavior. The stealth constraint is:

$$D_{\mathrm{KL}}\left(P_{\tilde{\nu}} \parallel P_{\nu}\right) \leq \epsilon \tag{2.13}$$

where $\epsilon > 0$ is a threshold on the permissible deviation. This ensures the attack remains within the statistical detection limits of the system.

The authors formulate this as a constrained optimization problem and solve it using convex programming tools based on the identified system behavior. Simulation results validate the proposed approach, showing that the attacker can substantially degrade the performance of the observer while remaining undetected, even without full system knowledge.

This work underscores the critical vulnerability of CPS to intelligent, data-driven adversaries and motivates the need for robust detection mechanisms that consider stealth constraints in the innovation domain. However, the stealthiness condition is defined in terms of the Kalman filter innovation and its distribution. This implicitly assumes that the defender is using a Kalman filter for state estimation. If alternative estimators (e.g., nonlinear filters or robust observers) are used, the attacker's stealth design may not be effective. These reviewed attacks use estimated models based on observed data. While these models may approximate system behavior well over a limited horizon, they are generally less accurate for long-term predictions or when the system operates in regions not well covered by the data. In this chapter, a novel covert data-driven attack is introduced that requires no system estimation, it will only use the data from the history of the system.

2.3 Problem Formulation

The covert-replay attack proposed in this chapter represents a potent threat, blending the characteristics of replay and covert attacks to undermine the performance of LTI systems. The main idea behind this attack is derived from both the covert attack and replay attack. This attack is designed to disrupt the optimal operation of the system while being stealthy from the command and control

perspective. It is predicated on the assumption that the targeted system is LTI, asymptotically stable, and operating in a steady state.

The objectives of this chapter are to (i) introduce and rigorously define the novel Data-Driven Covert-Replay Attack (DDCRA) that blends replay and covert strategies while requiring no model knowledge, (ii) analyze the attack's stealth mechanisms and limitations—in particular how it exploits linearity and superposition to preserve residual and watermark signatures—and quantify the trade-off between stealth and impact through the attacker gain m and timing parameters, (iii) demonstrate the ineffectiveness of standard additive watermarking against data-driven attacks and thereby motivate a new detection architecture, (iv) develop a model-based detection scheme that embeds a secret watermark inside the controller (while algebraically cancelling it before the actuator) so that tampering is revealed in the sensor measurements without degrading plant performance, and (v) validate the attack and the proposed detector in simulation using the quadruple-tank benchmark with an LQI controller and Kalman filter, reporting detection performance (false alarm and true positive rates) under realistic noise and disturbance. Together these objectives establish the threat model, expose weaknesses of existing defenses, and provide a practical, low-impact detection approach with quantitative evidence of its effectiveness.

The reliance on the LTI system condition primarily stems from leveraging the superposition principle inherent in such systems, which allows for the combination of multiple inputs to determine the system's output as

$$\begin{aligned} \textbf{For an input: } u_s &= k_1.u_1 + k_2.u_2 \\ \textbf{system output: } y_s &= k_1.y_1 + k_2.y_2 \end{aligned} \tag{2.14}$$

where y_i is the output of an LTI system for input u_i.

Consider the discrete-time linear system with process noise, measurement noise, and external disruptions is given by:

$$x(k+1) = Ax(k) + Bu(k) + w(k) \tag{2.15}$$

$$y(k) = Cx(k) + Du(k) + v(k) \tag{2.16}$$

where: $x(k) \in \mathbb{R}^n$ is the state vector at time step k, $u(k) \in \mathbb{R}^m$ is the control input, $y(k) \in \mathbb{R}^q$ is the measured output from the sensors, $w(k) \sim \mathcal{N}(0, Q)$ is the process noise (zero-mean Gaussian), $v(k) \sim \mathcal{N}(0, R)$ is the measurement noise (zero-mean Gaussian), A, B, C, D: are the system matrices of appropriate dimensions

This attack consists of 3 phases, each phase is a critical phase for the success of the attack.

2.3.1 Phase 1

In phase 1, this phase is similar to the replay attack; however, instead of only recording the output signal, the input is also recorded. Consider the attack starts at k_{p_1}, it records the input u and the output y for some time K_T. The period of K_T can vary, but it would not affect the attack. By the end of phase 1, the recorded input will be $u(k - K_T)$ and the recorded output is $y(k - K_T)$.

2.3.2 Phase 2

Phase 2 of this attack starts after the end of phase 1 at $k_{p_2} \geq k_{p_1} + K_T$. In this phase, the attacker aims to achieve the following stealthiness condition

$$\tilde{y}(k) = L(u(k)) \tag{2.17}$$

where $L(\cdot)$ denotes the system operator in discrete time (i.e., summation with the impulse response). The attacker will use the disruptive resources at the input and the output. The attack signal at the input will be

$$u_a = F(k).u(k - K_T) \tag{2.18}$$

which is the recorded input multiplied by gain

$$F(k) = \begin{cases} 0 & \text{for } k < k_{p_2} \\ \min\left(\dfrac{m \cdot (k - k_{p_2})}{k_s},\ f_s.m\right) & \text{for } k \geq k_{p_2} \end{cases} \tag{2.19}$$

where k_s is the sampling time, f_s is the saturating value of $F(k)$ and m is an attacker gain that will be critical in making the attack stealthy which will be discussed later in this chapter. Figure 2.2

shows an example of $F(k)$ signal.

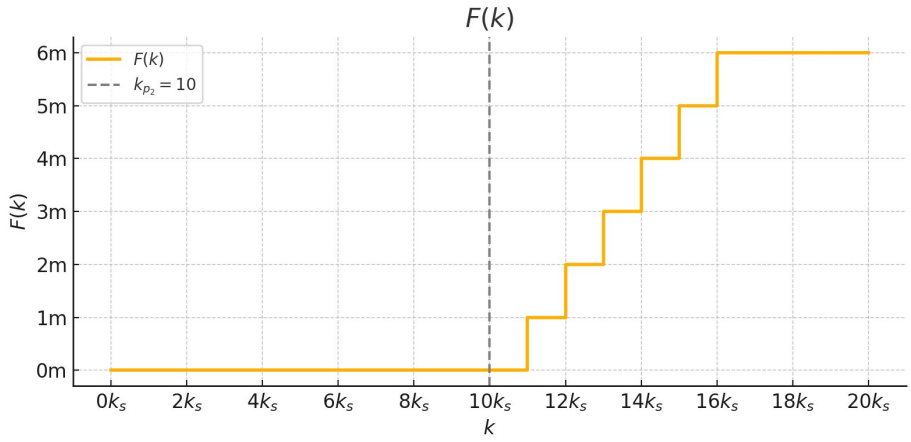

Figure 2.2: $F(k)$ with $f_s = 6$

Note that the attacker signal at the input u_a can only be a linear combination of the recorded input. This u_a will result in

$$\tilde{u} = u + f(k).u(k - K_T) \tag{2.20}$$

going to the plant causing the system to give an output

$$y = L(u) + L(f(k).u(k - K_T) \tag{2.21}$$

To implement a stealthy attack, we must use the disruptive resources at the output to design y_a that can cancel the effect caused by (2.18) and that would be

$$y_a = -[(f(k).y(k - K_T)] \tag{2.22}$$

The injected attack signal at the output will be the negative of the recorded output signal multiplied by $F(k)$. To prove the stealthiness, consider the following simple $F(k)$

$$f(k) = \begin{cases} m & 0 \leq k \leq k_1 \\ 2m & k > k_1 \end{cases} \tag{2.23}$$

Case I: $0 \leq k \leq k_1$

In this interval, the scaling function is constant: $f(k) = m$. The attacker injects the following signal from (2.18) at the input:

$$u_a(k) = m \cdot u(k - K_T) \tag{2.24}$$

The modified input received by the plant becomes:

$$\tilde{u}(k) = u(k) + m \cdot u(k - K_T) \tag{2.25}$$

Due to linearity of the system, the corresponding output is:

$$y(k) = L(u(k)) + L(m \cdot u(k - K_T)) \tag{2.26}$$

The system output becomes:

$$y(k) = \sum_{n=0}^{k_1} h(n)u(k - n) + m \sum_{n=0}^{k_1} h(n)u(k - K_T - n) \tag{2.27}$$

where $h(n)$ is the discrete-time impulse response of the system.

To remain covert, the attacker injects from (2.22):

$$y_a(k) = -m \cdot y(k - K_T) = -m \sum_{n=0}^{k_1} h(n)u(k - K_T - n) \tag{2.28}$$

Therefore, the final observed output becomes:

$$\tilde{y}(k) = y(k) - y_a(k) = \sum_{n=0}^{k_1} h(n)u(k - n) = L(u(k)) \tag{2.29}$$

and since $L(u(k - K_T)) = y(k - K_T)$ Thus, the attack achieves the stealth condition $\tilde{y}(k) = L(u(k))$ from Equation 2.17 during this interval.

Case II: $k > k_1$

In this interval, the scaling function is constant: $f(k) = 2m$. The attacker injects the following signal at the input:

$$u_a(k) = 2m \cdot u(k - K_T) \tag{2.30}$$

The modified input received by the plant becomes:

$$\tilde{u}(k) = u(k) + 2m \cdot u(k - K_T) \tag{2.31}$$

Due to linearity of the system, the corresponding output is:

$$y(k) = L(u(k)) + L(2m \cdot u(k - K_T)) \tag{2.32}$$

The system output becomes:

$$y(k) = \sum_{n=k_1}^{\infty} h(n)u(k-n) + 2m \sum_{n=k_1}^{\infty} h(n)u(k - K_T - n) \tag{2.33}$$

To remain stealthiness, the attacker injects:

$$y_a(k) = 2m \cdot y(k - K_T) = 2m \cdot L(u(k - K_T)) = 2m \cdot \sum_{n=0}^{k_1} h(n)u(k - K_T - n) \tag{2.34}$$

Therefore, the final observed output becomes:

$$\tilde{y}(k) = y(k) - y_a(k) = \sum_{n=k_1}^{\infty} h(n)u(k-n) \tag{2.35}$$

Thus, the attack achieves the stealth condition $\tilde{y}(k) = L(u(k))$ from Equation 2.17 during this interval.

It's worth noting that initially, the impact of this attack may seem insignificant because of the small value of m. However, over time, it can escalate to a critical level. The attacker's selection of the parameter m depends on several factors, including the duration available for the attack and

the associated risk. When ample time is available m can be set to a minimal value to significantly reduce the likelihood of detection. The smaller the m value the better, since you do not have any knowledge of the model and this will allow the attacker to observe the disruption in the system while increasing the value of $F(k)$ and control it to a desired level. However, a small value of m will require a long time to achieve the desired damage to the system achievable by the attacker

2.3.3 Phase 3

This is the relaxing stage of the attack and starts at $k_{p_3} = k_{p_2} + K_T$. The attack signal on the input becomes constant as

$$u_a = f(k_{p_3}).u(k - K_T) \tag{2.36}$$

which is the recorded output with constant gain $f(k_{p_3}) = f_s \cdot m$, this will keep the system at a new output

$$y = L(u(k)) + f_s \cdot m \cdot L(u(k - K_T)) \tag{2.37}$$

and by injecting

$$y_a = -[f_s \cdot m \cdot .y(k - K_T)] \tag{2.38}$$

the output seen by the monitor would adhere to the stealth condition provided by Equation 2.17. The block diagram of the attack is shown in Figure 2.3. The three stages of the output signal of covert attack are shown in Figure 2.4. Note that Phase 1 has to be recorded during steady state.

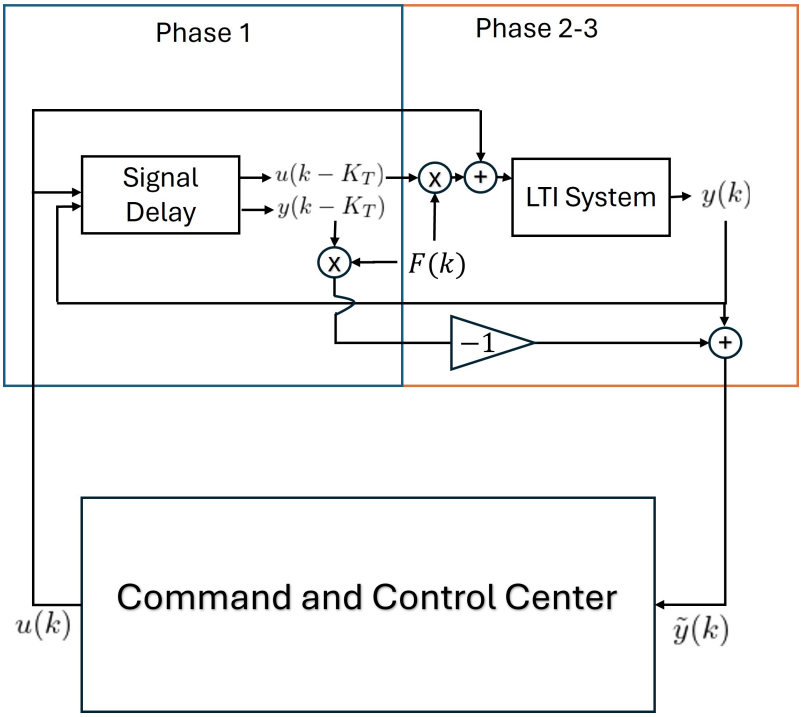

Figure 2.3: Data-Driven Covert Replay Attack Block Diagram

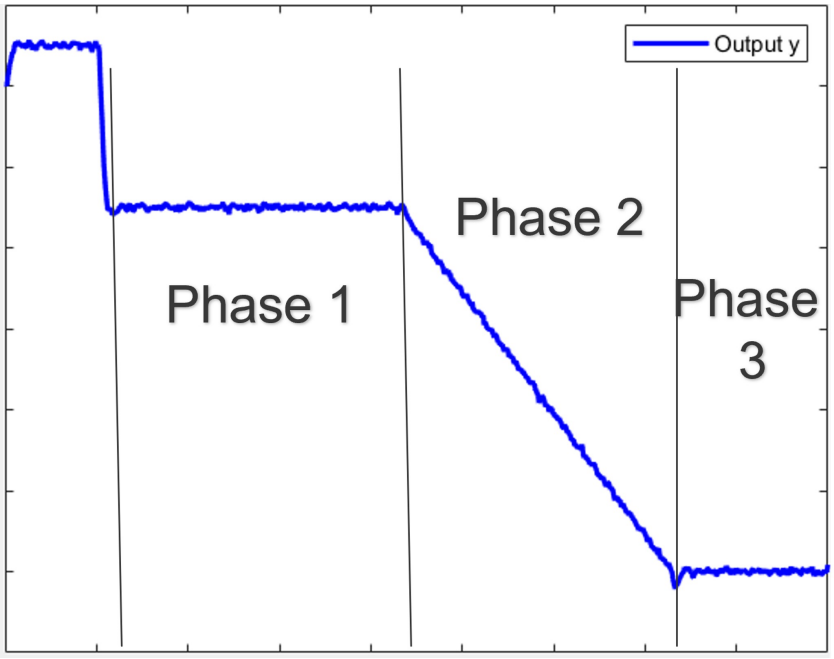

Figure 2.4: Data-Driven Covert Replay Attack Output Stages

2.3.4 Limitation of Additive Watermarking Against Covert-Replay Attack

Additive watermarking is a commonly employed data-based detection technique, where a known random signal is injected into the control input to verify integrity via correlation with the observed system output. Specifically, the control input is modified as $u(k) = u_c(k) + \omega(k)$, where $\omega(k)$ is a zero-mean, statistically independent watermark known only to the defender. In normal operation, the output $y(k)$ maintains statistical correlation with the injected watermark due to the system dynamics. Detection algorithms monitor this correlation to identify malicious interference.

However, the covert-replay attack described in this work inherently preserves the watermark signal, rendering such detection ineffective. During Phase 1 of the attack, the adversary records both the input $u(k)$ and the corresponding output $y(k)$, which implicitly includes the watermark $\omega(k)$. In subsequent attack phases, the attacker injects scaled versions of the recorded input and output—specifically, $u_a(k) = f(k) \cdot u(k - K_T)$ and $y_a(k) = -f(k) \cdot y(k - K_T)$—ensuring that the watermark's statistical imprint is preserved.

As a result, the observed output during the attack becomes $\tilde{y}(k) = y(k) - y_a(k) = L(u(k))$, which is indistinguishable from the normal system response. Since the watermark $\omega(k)$ remains embedded in both the replayed input and output, the correlation-based detection metric remains unaltered, thereby failing to raise an alarm.

2.4 Model-Based Detection

To enhance stealth attack detection while mitigating performance degradation, we propose a detection method inspired by additive watermarking, wherein a known excitation signal is embedded into the control loop. Unlike traditional watermarking, the key distinction of this approach lies in the cancellation of the watermark signal before it reaches the physical actuator. Specifically, the watermark is introduced after the controller, yet it is removed prior to signal transmission to the plant.

This design preserves the system's nominal performance and avoids contaminating the plant dynamics with unnecessary excitation, while still allowing the detector to evaluate whether the expected signature of the watermark is present in the sensor measurements. Absence or distortion of

this signature may indicate tampering, thus enabling attack detection without compromising system stability or output quality.

The signal is going to be sent from the command and control center will be

$$u = u^*(k) + \omega(k) \tag{2.39}$$

where u^* is the optimal control output. The recorded signal from the attacker in phase one will be $u(k - K_T)$. However, the defender will remove the effect of this at the plant side ensuring u^* to be the signal received by the actuators. This requires that the defender have a secret seed to the watermark signal that can be replicated at the plant side. Since the attacker in phase 1 will record the output simultaneously which is $y(k - K_T) = L(u^*(k - K_T)$, in phase 2 the injected signals from 2.18 and 2.22 will be

$$u_a = F(k).(u^*(k - K_T) + \omega(k - K_T)) \tag{2.40}$$

and

$$y_a = F(k).(y(k - K_T)) \tag{2.41}$$

, considering the example of $F(k)$ in (2.23), for $0 \leq k \leq k_1$, the output from $\tilde{u} = u^* + u_a$ will be

$$L(\tilde{u})(k) = L(u^*(k)) + m.L(u^*(k - K_T)) + m.L(\omega(k - K_T) \tag{2.42}$$

. The result after injecting y_a as in (2.29) will be

$$\tilde{y}(k) = L(u^*(k)) + m.L(u^*(k - K_T)) + m.L(\omega(k - K_T) - m.y(k - K_T) \tag{2.43}$$

and since $m.L(u^*(k - K_T)) = m.y(k - K_T)$, the final result is

$$\tilde{y}(k) = L(u^*(k)) + m.L(\omega(k - K_T)) \tag{2.44}$$

which is not stealthy according to the stealthy condition in (2.17).

2.5 Simulation Results

2.5.1 Quadruple-Tank Process

To evaluate the proposed detection framework, we adopt the well-established *quadruple-tank process* as the simulation environment. Originally introduced by [71], this process is widely used in control systems research due to its multivariable dynamics and ability to exhibit both minimum-phase and non-minimum-phase behaviors.

The quadruple-tank system consists of four interconnected water tanks and two pumps. The control inputs to the system are the voltages applied to the pumps, denoted u_1 and u_2, which regulate the inflow rates. The outputs of interest are the water levels in the two lower tanks, denoted y_1 and y_2. The coupling between tanks is configured through a set of valves that distribute the flow between upper and lower tanks, allowing the system to simulate various multivariable interactions as shown in Figure 2.5.

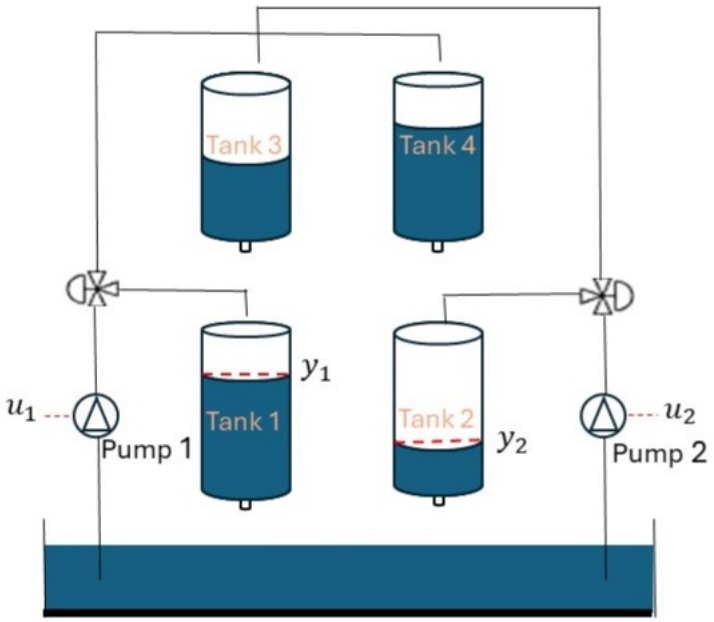

Figure 2.5: Quadruple Water Tank System

The control inputs $u = [u_1, u_2]^\top$ are constrained within the physical pump voltage range:

$$u_i \in [0, 10] \quad \text{(Volts)}, \quad i = 1, 2$$

and the output water levels $y = [y_1, y_2]^\top$ are measured in centimeters with operating bounds:

$$y_i \in [0, 25] \quad \text{(cm)}, \quad i = 1, 2$$

This bounded coupled system provides a realistic testbed for evaluating both prediction accuracy and anomaly detection under CPS attacks.

The quadruple-tank process is a laboratory setup consiting of four interconnected water tanks and two pumps, designed to illustrate control systems problems such as nonminimum-phase behavior [71]. The system enables the adjustment of zero between the two half-planes by controlling the pumps.

Each tank $i \in \{1, 2, 3, 4\}$ has a cross-sectional area A_i, and each outlet orifice has an area a_i. The liquid levels in the tanks are denoted $h_i(t)$ in centimeters. The two pumps has flow rates proportional to their respective voltages $v_1(t)$ and $v_2(t)$ input, represented as k_1 and k_2, respectively. The flow from each pump is divided by adjustable valves that determine the fraction $\gamma_1, \gamma_2 \in [0, 1]$ directed toward the lower tanks (Tanks 1 and 2), while the remaining fractions $(1-\gamma_1)$ and $(1-\gamma_2)$ are fed to the upper tanks (Tanks 4 and 3), respectively. The acceleration of gravity is denoted $g = 981$ cm/s^2.

The nonlinear dynamics of the quadruple-tank system are governed by mass balance and Bernoulli's law:

$$\dot{h}_1 = -\frac{a_1}{A_1}\sqrt{2gh_1} + \frac{a_3}{A_1}\sqrt{2gh_3} + \frac{\gamma_1 k_1}{A_1}v_1, \tag{2.45}$$

$$\dot{h}_2 = -\frac{a_2}{A_2}\sqrt{2gh_2} + \frac{a_4}{A_2}\sqrt{2gh_4} + \frac{\gamma_2 k_2}{A_2}v_2, \tag{2.46}$$

$$\dot{h}_3 = -\frac{a_3}{A_3}\sqrt{2gh_3} + \frac{(1-\gamma_2)k_2}{A_3}v_2, \tag{2.47}$$

$$\dot{h}_4 = -\frac{a_4}{A_4}\sqrt{2gh_4} + \frac{(1-\gamma_1)k_1}{A_4}v_1. \tag{2.48}$$

The measurable outputs are the levels in the lower tanks:

$$y(t) = \begin{bmatrix} h_1(t) \\ \\ h_2(t) \end{bmatrix}.$$

Let h_{i0} and v_{j0} denote the steady-state water levels and pump voltages, respectively, corresponding to an equilibrium point. Linearizing (2.45)–(2.48) around the operating point results in

$$\dot{x} = Ax + Bu, \qquad y = Cx,$$

where the state and input vectors are defined as

$$x = \begin{bmatrix} h_1 - h_{10} \\ h_2 - h_{20} \\ h_3 - h_{30} \\ h_4 - h_{40} \end{bmatrix}, \quad u = \begin{bmatrix} v_1 - v_{10} \\ \\ v_2 - v_{20} \end{bmatrix}.$$

The linearized system matrices take the form

$$A = \begin{bmatrix} -\frac{1}{T_1} & 0 & \frac{A_3}{A_1 T_3} & 0 \\ 0 & -\frac{1}{T_2} & 0 & \frac{A_4}{A_2 T_4} \\ 0 & 0 & -\frac{1}{T_3} & 0 \\ 0 & 0 & 0 & -\frac{1}{T_4} \end{bmatrix}, \quad B = \begin{bmatrix} \frac{\gamma_1 k_1}{A_1} & 0 \\ 0 & \frac{\gamma_2 k_2}{A_2} \\ 0 & \frac{(1-\gamma_2) k_2}{A_3} \\ \frac{(1-\gamma_1) k_1}{A_4} & 0 \end{bmatrix},$$

and the output matrix

$$C = \begin{bmatrix} 1 & 0 & 0 & 0 \\ \\ 0 & 1 & 0 & 0 \end{bmatrix}.$$

The parameters $T_i = \frac{A_i}{a_i}\sqrt{\frac{2h_{i0}}{g}}$ represent the time constants of the individual tanks.

Using the nominal parameters reported by [71],

$$A_1 = A_3 = 28\,\text{cm}^2, \quad A_2 = A_4 = 32\,\text{cm}^2, \quad a_1 = a_3 = 0.071\,\text{cm}^2, \quad a_2 = a_4 = 0.057\,\text{cm}^2, \quad g = 981\,\text{cm/s}^2,$$

and assuming $k_1 = 3.33,\ k_2 = 3.35$, the linearized continuous-time models for the two operating conditions are given below.

(a) Minimum-Phase Configuration For $(\gamma_1, \gamma_2) = (0.70,\ 0.60)$ and steady-state levels $(h_{10}, h_{20}, h_{30}, h_{40}) = (12.4,\ 12.7,\ 1.8,\ 1.4)$:

$$A_{P^-} = \begin{bmatrix} -0.0159 & 0 & 0.0419 & 0 \\ 0 & -0.0111 & 0 & 0.0333 \\ 0 & 0 & -0.0419 & 0 \\ 0 & 0 & 0 & -0.0333 \end{bmatrix}, \quad B_{P^-} = \begin{bmatrix} 0.0833 & 0 \\ 0 & 0.0628 \\ 0 & 0.0479 \\ 0.0312 & 0 \end{bmatrix}.$$

(b) Nonminimum-Phase Configuration For $(\gamma_1, \gamma_2) = (0.43,\ 0.34)$ and steady-state levels $(h_{10}, h_{20}, h_{30}, h_{40}) = (12.6,\ 13.0,\ 4.8,\ 4.9)$:

$$A_{P^+} = \begin{bmatrix} -0.0158 & 0 & 0.0256 & 0 \\ 0 & -0.0109 & 0 & 0.0178 \\ 0 & 0 & -0.0256 & 0 \\ 0 & 0 & 0 & -0.0178 \end{bmatrix}, \quad B_{P^+} = \begin{bmatrix} 0.0482 & 0 \\ 0 & 0.0350 \\ 0 & 0.0776 \\ 0.0559 & 0 \end{bmatrix}.$$

These 4×4 and 4×2 matrices fully characterize the linearized quadruple-tank system and provide a complete numerical description suitable for simulation, controller design, and comparative analysis of minimum- and nonminimum-phase behaviors.

An LQI controller with a kalman filter is designed to follow this reference signal

$$y_{ref}(k) = [15, 10]^\top$$

and as shown in Figure 2.6, The LQI controller is working properly since a transient response

existed from $t = 0$ untill around $t = 300$ and then the reference signal is tracked perfectly. It can be noticed that there is no fluctuations in the outputs since this simulation has no noise or disturbance for simplicity.

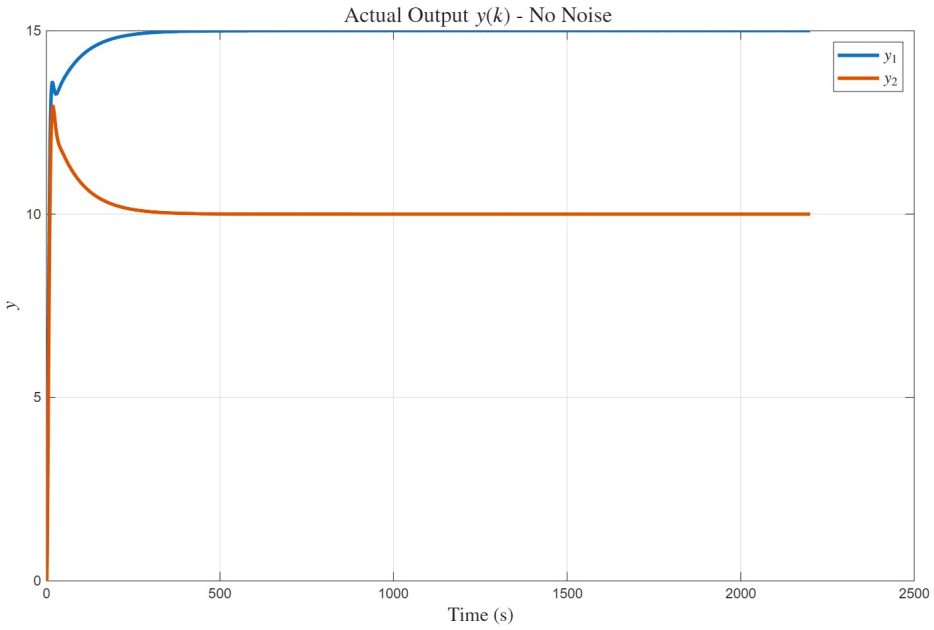

Figure 2.6: LQI Controller Tracking $y_{ref}(k)$ without noise or disturbance

The data-driven covert attack is introduced with $f_s = 500$ and $m = 0.0005$. The stage 1 of the attack starts at $k_{p1} = 700$, and the recording duration $K_T = 500$. The stage 2 starts at $k_{p2} = 1200$, and stage 3 starts at $k_{p3} = 1700$. Firstly, the attack is going to be studied without noise or disturbance ($w(k) = 0$ and $v(k) = 0$). This is just to observe the results of the attack in a simpler way visually. The attack results without noise is shown in Figure 2.7, As can be observed, in the first stage of the attack between $t = 700$ and $t = 1200$, the actual y values are not being altered as it is just being recorded and saved in memory to be used in later stages. However, during the second stage between $t = 1200$ and $t = 1700$, the output values y are increasing in a steady way depending on how $f(k)$ is increased. In this case, it was increased in a constant form similar to Figure 2.2, during this stage the $\tilde{y}$ deviated a bit, and this happens when m values are big compared the open loop response of the system as will be analyzed in details in Figure 2.8. Nonetheless,

when it reached the third stage at $t = 1700$, the output $y(k)$ is altered to a value of $y = [13, 19]^\top$, however, the $\tilde{y}(k)$ is exactly similar to the Figure 2.6 which is the operating LQI controller without an attack, which is completely stealthy to the command and controller.

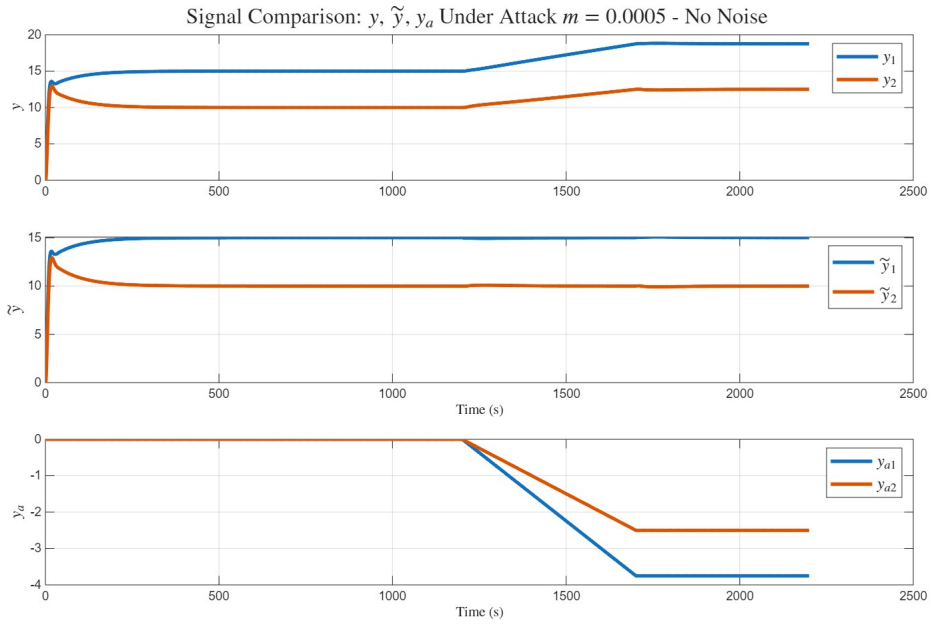

Figure 2.7: The Output Signals and Attack Signal on the output channels results

The second stage $t = (1200, 1700)$ is mainly affected by the function $f(k)$ which has the gain m. Different m values can affect the sharpness of the attack as shown in Figure 2.8. That is because the transient response of the system is triggered once the attack phase 2 starts untill the start of phase 3, and the transient response of the system could be sharp since it depends on the open loop response of the original system. To minimize this, lower values of m will keep $\tilde{y}(k)$ during this stage as close as possible to $y_{ref}(k)$ which is the main goal to be stealthy. In Figure 2.8, During stage 2 for $m = 0.001$,the values of $\tilde{y}$ deviated, which a normal $||y - C\hat{x}||$ residual based detector can detect it assuming no noise. This is mainly because the m value is considered a little high for this system.

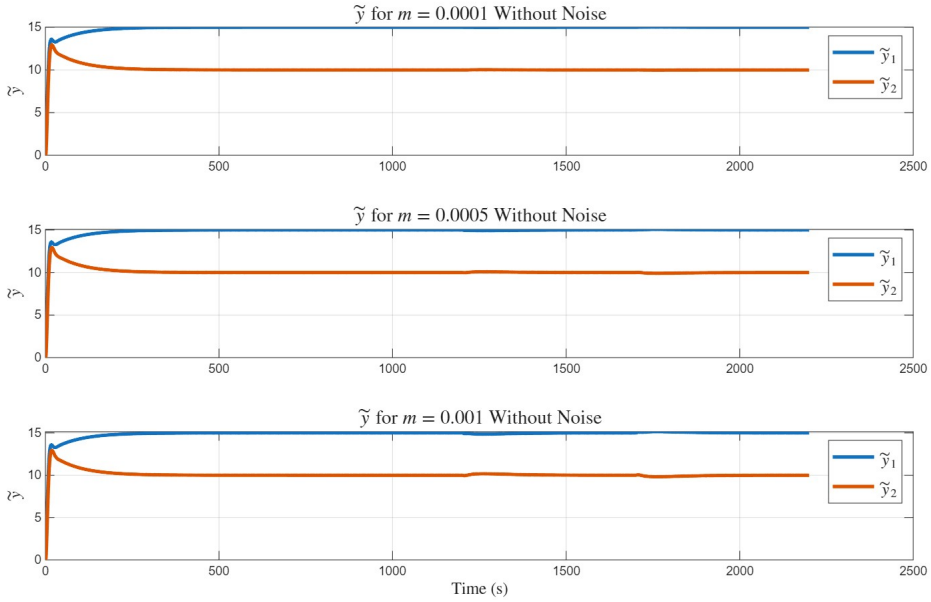

Figure 2.8: The $\tilde{y}(k)$ for different m values under no noise

However, for $m = 0.0001$ the attack is much smoother and essentially undetectable since $\tilde{y}(k) = y_{\text{ref}}(k)$; however, the actual plant output y deviates far less for the same Stage-2 duration: for $m = 0.001$ the output moves from $[15,\ 10]^\top$ to approximately $[22,\ 15]^\top$, whereas for $m = 0.0001$ it shifts only from $[15,\ 10]^\top$ to about $[16,\ 11]^\top$ (see Fig. 2.9). So for the attacker to control y to reach a certain desired level that is far from the $y_{ref}(k)$ with a lower value of m, stage 2 will require drastically more time since the change of y values is tied to m values, and the m values are tied to the stealthiness of the attack.

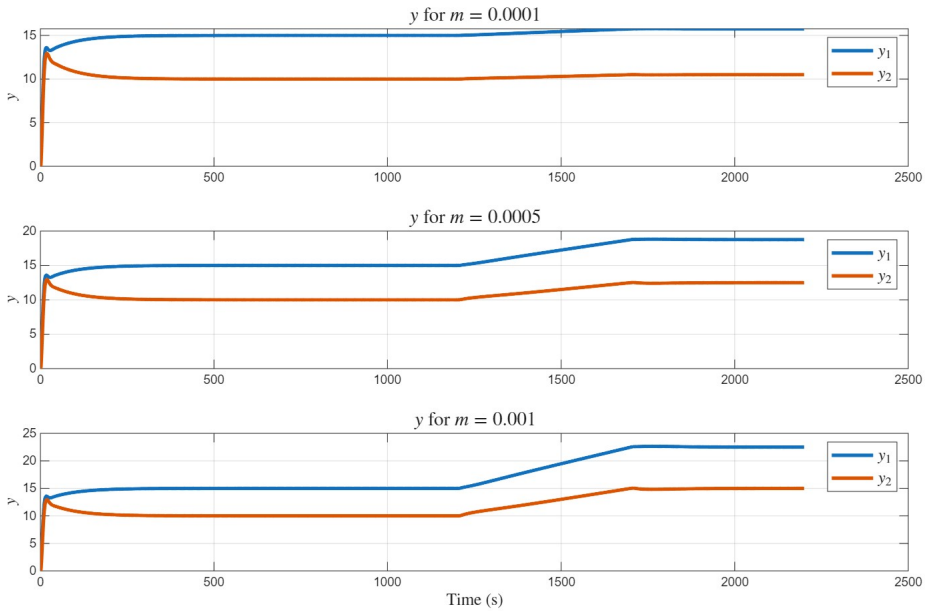

Figure 2.9: The Output Signals and Attack Signal on the Output Channels

Now, the disturbance is introduced as following, $w(k) \sim \mathcal{N}(0,\ 0.01^2)$ and measurement noise as $v(k) \sim \mathcal{N}(0,\ 0.04^2)$. The output with the absence of attacks is shown in Figure 2.10. This figure is exactly similar to Figure 2.6 but with constant deviations since the only difference is the disturbances and noises added, which verifies the desired operating conditions under the noise and disturbance.

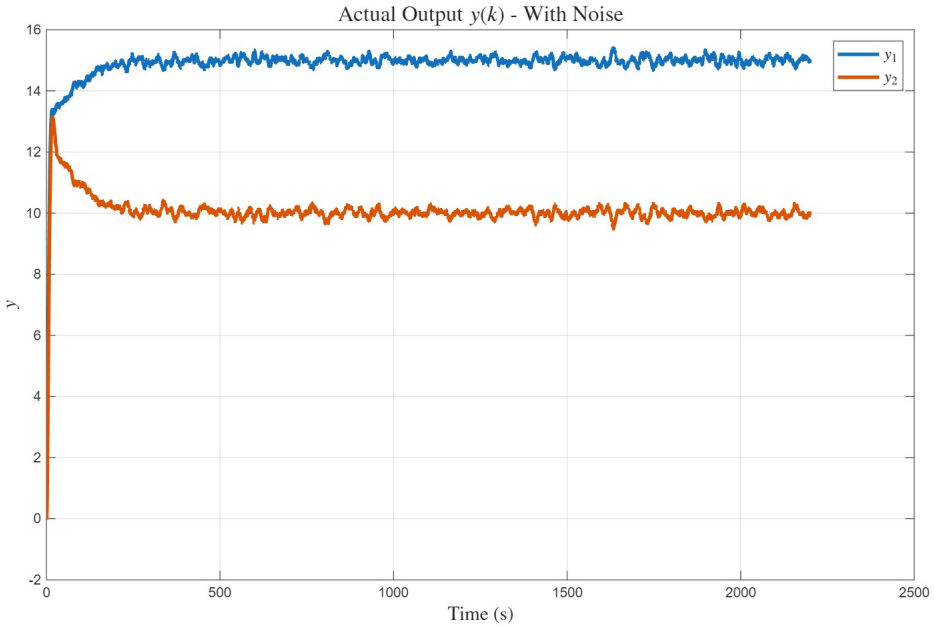

Figure 2.10: Controlled System Output With Noise

The noise and disturbance introduce the stochastic problem in detecting attacks which includes balancing the false alarm rate under no attack to the true positive rate during attacks. However, the system figures to the human eye just look the same as without noise, as shown in Figure 2.11. The $\tilde{y}$ is what the controller see, and in Figure 2.11 the value of $\tilde{y}$ remains unchanged to the expected reference even with the noise introduced.

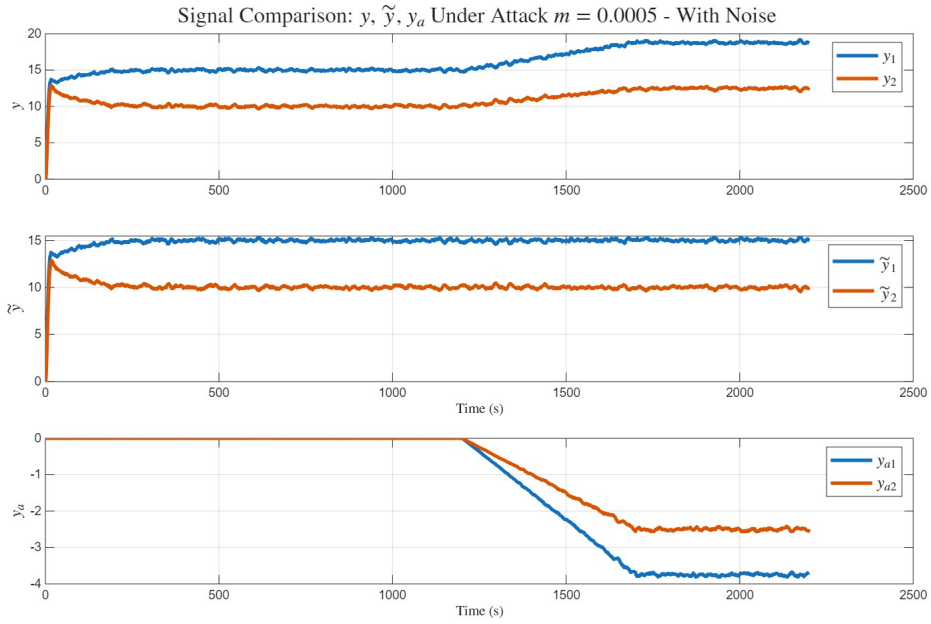

Figure 2.11: Data-Driven Covert Attack Output Stages

However, visually it is hard to keep track of the detectors ability to detect such attacks, that is why a stochastic detection approach must be introduced to test the results as in the next section.

2.5.2 Chi-Square Residual-Based Detection and Watermark

To evaluate the stealthiness of the proposed DDCRA, a residual-based anomaly detection scheme was implemented using the chi-square test, following the methodology presented in [13].

The residual-based detector is designed to identify deviations between the measured outputs and the predicted outputs from the estimator. Let $\hat{x}$ denote the estimated state vector and y the measured output vector. The detector proceeds as follows:

(1) **Residual Calculation:** The residual vector r is defined as the difference between the measured output and the predicted output:

$$r = y - C\hat{x}, \tag{2.49}$$

where C is the output matrix selecting the measured states.

(2) **Residual Covariance:** The covariance of the residual, S, is computed using the kalman filter covariance P and the measurement noise covariance R:

$$S = CPC^\top + R. \tag{2.50}$$

(3) **Test Statistic:** The detector evaluates the Mahalanobis distance of the residual:

$$z = r^\top S^{-1} r. \tag{2.51}$$

(4) **Threshold Comparison:** A chi-square threshold τ is used to decide whether an attack is present. For a desired confidence level α and degrees of freedom equal to the number of measured outputs n_y, the threshold is

$$\tau = \chi^2_{\alpha, n_y}. \tag{2.52}$$

The detection decision is then

$$\text{Attack detected} = \begin{cases} \text{True}, & z > \tau, \\ \text{False}, & z \leq \tau. \end{cases} \tag{2.53}$$

This formulation allows the detector to systematically identify anomalies in the system outputs by comparing the residual's deviation against the statistical threshold derived from the estimator and sensor noise characteristics.

The detector was designed to maintain a 5% false alarm rate under normal operating conditions.

This detector monitors the residual signal generated by the Kalman filter and raises an alarm whenever the deviation exceeds the chi-square threshold corresponding to the 95th percentile. The threshold is determined based on the output dimension of the system and the residual covariance.

Watermark detection method mainly implemented against data-driven attacks (replay attacks) from [72], is formed by adding a small random "watermark" signal to the control input. Let

$u_{\text{nominal}}(k)$ be the nominal control input at time k, then the actual input with watermark is

$$u(k) = u_{\text{nominal}}(k) + w(k),$$

where $w(k)$ is a low-amplitude, statistically known signal (e.g., Gaussian noise). The system residual

$$r(k) = y(k) - C\hat{x}(k),$$

is monitored using chi-square as explained previously. A replay attack is detected if $z(k)$ exceeds a threshold τ based on the chi-square distribution. This approach can be implemented in real time by generating the watermark sequence, adding it to the control input, and continuously monitoring the residual for deviations indicative of replayed signals.

The chi-square detector was tested under the presence of noise ($w(k) \sim \mathcal{N}(0, 0.01^2)$, $v(k) \sim \mathcal{N}(0, 0.04^2)$) for different values of the attacker gain m. The observed alarm rate, defined as the percentage of time steps where an alarm was raised, is presented in Table 2.1 with watermark defense system [72]. The alarm rate is supposed to be 5% during normal operation, and the tables below present the alarm rate percentage during the attacks. The results were the average of 50 simulation runs.

Table 2.1: Alarm Rate vs. Attacker Gain m

Gain m	**Alarm Rate (%)**		
	Chi-Square	**With Watermark**	**Model-Based Detection from 2.4**
0.0005	4.8	5.1	84
0.001	5.2	5.3	87
0.002	7.1	7.8	88
0.005	38.4	55.3	90
0.01	57.2	77.7	93

Table 2.2: F1 Score vs. Attacker Gain m

Gain m	**F1 Score (%)**		
	Chi-Square	**With Watermark**	**Model-Based Detection from 2.4**
0.0005	6.7	6.8	83
0.001	7.4	7.9	86
0.002	8.2	8.7	88
0.005	50.6	68.6	90
0.01	77.2	85.7	93

As shown in Table 2.1 and Table 2.2, with watermark and chi-square detection, the attack remains stealthy for small values of m, with alarm rates close to the nominal false alarm rate (5%) and a poor F1 score. This shows that the attack can be implemented successfully against those detectors as long as the attacker maintain a low m value. However, as the value of m increases, the deviation in the residual signal becomes more detectable by the traditional detectors. This validates the theoretical trade-off between stealth and time available to construct the attack. Also, the proposed detection method works effectively for all values of m as shown by the high alarm rate during the attack with an effective F1 score.

2.5.3 Comparison Between the Proposed Attack and Existing Approaches

The proposed **DDCRA** introduces a hybrid threat that blends elements of replay and covert attacks. Table 2.3 provides a structured comparison across key dimensions with notable data-driven and model-based attacks in the literature.

Table 2.3: Comparison Between DDCRA and Representative Attacks

Feature	**Replay Attack** [48]	**Covert Attack** [57]	**Self-Generated FDI Attack** [47]	**Proposed DDCRA**
Model Knowledge Requirement	None	Full model knowledge (A, B, C)	Estimated via subspace ID	**None**
Attacker's Data Requirement	Recorded outputs $y(k)$	Full model dynamics $G(s)$	Historical I/O data	Recorded $u(k), y(k)$
Actuator Interference	None	Injected $u_a(k)$	No (sensor-side only)	Injected $u_a(k) = f(k) \cdot u(k - K_T)$
Sensor Interference	Replayed $\tilde{y}(k)$	Injected $y_a(k) = G(s)u_a(s)$	Injected $y_a(k) = Ce_a(k)$	Injected $y_a(k) = -f(k) \cdot y(k - K_T)$
Stealth Strategy	Replaying valid sensor data	Output cancellation via model inversion	Mimic residual dynamics	Cancel impulse response via superposition
Detection Evasion	Residual statistics preserved	Residual unchanged by design	Innovation sequence unchanged	Preserves both residual and watermark signals
Observer/Estimator Knowledge	Not required	Required	Not Required	**Not required**
Robustness to Noise	Moderate	High (model accurate)	Sensitive to modeling error	**Robust (validated with Gaussian noise)**

Discussion: Unlike classical replay attacks, the DDCRA introduces simultaneous manipulation on both input and output channels and leverages linear system superposition for stealth. Compared to covert attacks, it eliminates the need for full or partial model knowledge. Unlike self-generated FDI attacks, it does not require Kalman filter design or subspace identification, making it *truly model-free*. Moreover, DDCRA can bypass additive watermark-based detection by preserving the watermark's signature in recorded signals. The main advantage this attack has over [47] is the requirement of kalman filter knowledge in order to bypass the detector. The advantage of DDCRA over other covert attacks such as [57] in the literature is the absence of the requirement to estimate the system matrices. Although system identification is possible, before launching a model-based attack, an adversary often needs to perform system identification to learn how the physical process behaves. However, this step comes with several drawbacks. To build an accurate model, the attacker

must collect enough rich input–output data, which can be difficult without drawing attention or disrupting normal operations.

2.6 Chapter Summary

This chapter presented the DDCRA, an innovative stealthy cyber-attack that combines the operational principles of covert and replay attacks to infiltrate LTI control systems without needing model knowledge. The attack takes advantage of the superposition principle of LTI systems to give the command and control center the illusion of normal functioning. The DDCRA occurs in three consecutive stages: (i) The data collection stage, in which the control input and observed output are recorded while in steady-state operation; (ii) The injection stage, where scaled copies of the recorded signals are injected into the sensors and actuators using a gain function $F(k)$; and (iii) a relaxation stage, during which the attack maintains system deviation in steady conditions.

Analytical derivations demonstrated how the injected input $u_a(k) = F(k)\, u(k - K_T)$ and output compensation $y_a(k) = -F(k)\, y(k - K_T)$ satisfy the stealthiness condition $\tilde{y}(k) = L(u(k))$, thereby the output has consistent residual bypassing any residual based detector. The attacker gain m and saturation level f_s were shown to control the trade-off between attack impact and detectability—small values of m yielding higher stealthiness but with slower disruption.

Furthermore, this chapter established the limitations of conventional additive watermarking techniques against DDCRA. Since the attacker replays previously recorded input–output pairs that already contain the embedded watermark, the correlation-based detector fails to distinguish the attack from legitimate system behavior. This vulnerability underscores the necessity for more advanced detection architectures capable of identifying data-driven stealth attacks while preserving system performance.

Moreover, this chapter showed the constraints of traditional additive watermarking methods in relation to DDCRA. As the attacker is able to use recorded input-output pairs that have the embedded watermark, the correlation-based detector cannot differentiate between the attack and valid system operation because the effect of watermarking being injected into the system is removed at the output due to the covert attack.

Chapter 3

Reinforcement Learning for Cyber-Attack Detection in CPS

CPS are progressively more susceptible to a variety of cunning and covert assaults, as shown in earlier chapters.Conventional model-based and data-oriented detection methods encounter significant drawbacks in adjusting to rapidly changing threats.Model-based approaches frequently demand precise understanding of the system, whereas data-driven methods are vulnerable to adversarial attacks or may fail to generalize. Conversely, **Reinforcement Learning (RL)** has developed into an encouraging framework that allows CPS to independently adjust and discover optimal detection strategies by interacting with their surroundings.This chapter explores the use of RL in protecting CPS, offering a targeted literature review and a systematic examination of RL-driven anomaly detection methods.

3.1 Background on Reinforcement Learning

Reinforcement Learning is about learning via trial and error, where an agent observes a state, performs an action, and receives feedback as a reward while interacting with an environment. The agent aims to discover a policy that maximizes the total expected reward throughout time.In CPS security, the environment denotes the system being monitored and the communication network linked to the command and control center, while the agent signifies the detection system.Potential actions

could include activating alarms, isolating parts, or improving detection models.RL can be categorized into various algorithms: value-focused techniques such as Q-learning and SARSA, policy-based methods including Policy Gradient and Proximal Policy Optimization (PPO), and combined approaches like Actor-Critic, Deep Q-Networks (DQN), and Deep Deterministic Policy Gradient (DDPG).These various techniques enable RL agents to operate in extensive state spaces with differing levels of observability.The properties of the action and observation spaces influence the design of RL algorithms.In discrete action-observation environments, the possible actions and the states that can be observed are either finite or countably infinite. This situation is common in traditional control problems.For instance, Q-learning utilizing discrete features or DQN where the states are continuous but the action space remains discrete. Conversely, in continuous action-observation spaces, the state and/or action dimensions are uncountably infinite and are often depicted as vectors with real numbers. This segment emphasizes two important and essential deep RL algorithms—DQN and DDPG—that have proven successful in different control environments. Their core design principles establish a basis for numerous detection systems in CPS.

3.1.1 Deep Q-Network

The Deep Q-Network (DQN) algorithm, first introduced by [73], is a foundational deep reinforcement learning approach that combines Q-learning with deep neural networks. DQN approximates the optimal action-value function $Q(s, a)$ using a deep neural network parameterized by θ. At each time step i within an episode, the agent observes a transition tuple $b_i = (s_i, a_i, s_{i+1}, r_i)$ and updates the Q-network by minimizing the mean-squared Bellman error defined as:

$$L(\theta) = \mathbb{E}_{(s_i, a_i, r_i, s_{i+1}) \sim R} \left[\left(y_i - Q(s_i, a_i; \theta) \right)^2 \right] \tag{3.1}$$

where the target value y_i is given by:

$$y_i = r_i + \gamma \cdot \max_{a'} Q'(s_{i+1}, a'; \theta^-) \tag{3.2}$$

In this formulation, Q' is the target network parameterized by θ^-, which is a delayed copy of the Q-network used to stabilize training. The target network is periodically updated as:

$$\theta^- \leftarrow \theta \tag{3.3}$$

To further enhance stability and reduce correlations in sequential observations, DQN employs a replay buffer R, which stores past transitions and samples mini-batches for training. The algorithm also utilizes an ϵ-greedy policy to balance exploration and exploitation, where the agent selects a random action with probability ϵ and the greedy action $\arg\max_a Q(s, a; \theta)$ otherwise. The DQN algorithm is summarized in Algorithm 3.

Algorithm 1 DQN Method

Initialize Q-network $Q(s, a; \theta)$ with random weights θ
Initialize target network $Q'(s, a; \theta^-)$ with $\theta^- \leftarrow \theta$
Initialize replay buffer R
for episode = 1 to M **do**
    Initialize state s_0
    for t = 1 to T **do**
        With probability ϵ select a random action a_t, otherwise $a_t = \arg\max_a Q(s_t, a; \theta)$
        Execute a_t, observe reward r_t and next state s_{t+1}
        Store transition (s_t, a_t, r_t, s_{t+1}) in R
        Sample a mini-batch of N transitions (s_i, a_i, r_i, s_{i+1}) from R
        Compute target y_i using Equation 3.2
        Update Q-network by minimizing loss in Equation 3.1
        Every C steps, update target network: $\theta^- \leftarrow \theta$ using (3.3)
    end for
end for

3.1.2 Deep Deterministic Policy Gradient

The DDPG approach was originally developed in [74], which is a deep RL algorithm that outputs a continuous control and has achieved great success in many different simulations. DDPG approximates the state action value function by using a critic function Q parameterized by θ^Q. It outputs the continuous action control given a current observation by the actor μ parameterized by θ^μ. Let $b_i = (s_i, a_i, s_{i+1}, r_i)$ exist for each time step in an episode in which a reward r_i is received after moving to state s_{i+1} caused by action a_i at the state s_i for $i = [1, 2, \ldots, N]$, where N is the

total number of batches b_i. The critic is updated by minimizing the loss function

$$L = \frac{1}{N}\sum_{i=1}^{N}(y_i - Q(s_i, a_i|\theta^Q))^2 \tag{3.4}$$

in which

$$y_i = r_i + \gamma.Q'(s_{i+1}, \mu'(s_{i+1}|\theta^{\mu'})|\theta^{Q'}) \tag{3.5}$$

where target networks μ' and Q' parameterized by $\theta^{\mu'}$ and $\theta^{Q'}$ are made to reduce the tracking speed of the original networks where it is updated as

$$\theta' \leftarrow \tau.\theta + (1-\tau).\theta' \tag{3.6}$$

with $\tau << 1$. The sampled policy gradient is used to update the actor policy μ as

$$\nabla_{\theta^\mu} J \approx \frac{1}{N}\sum_{i} \nabla_{\mu(s_i)} Q(s_i, \mu(s_i)|\theta^Q) \nabla_{\theta^\mu} \mu(s_i|\theta^\mu) \tag{3.7}$$

and the exploration policy μ' which is considered a challenge in continuous action spaces is constructed by adding a sampled noise from a noise process $\mathcal{N}$ to the actor policy

$$\mu'(s_i) = \mu'(s_i|\theta_i^\mu) + \mathcal{N} \tag{3.8}$$

where $\mathcal{N}$ is chosen as an Ornstein-Uhlenbeck process for exploration efficiency in physical control problems with inertia by producing correlated explorations. The DDPG algorithm is summarized in Algorithm 2.

3.2 Literature Review of RL-Based Detection Systems

Recent studies have demonstrated the applicability of RL to CPS anomaly detection with varying architectural designs, reward formulations, and system contexts.

A Double Deep Q-Network (DDQN) was developed by [75] to classify system states in CPS as

Algorithm 2 DDPG Method

Initialize Critic Network $Q(s_i, \mu(s_i)|\theta^Q)$ and Actor Network $\mu(s_i|\theta^\mu)$ with random θ^Q and θ^μ
Initialize target networks μ' and Q' with weights $\theta^{\mu'} \leftarrow \theta^\mu$ and $\theta^{Q'} \leftarrow \theta^Q$
Initialize Replay Buffer R
for episode = 1 to M **do**
    Initialize a random $\mathcal{N}$
    Receive initial state s_0
    for t = 1 to T **do**
        select action $a_t = \mu'(s_t|\theta_i^\mu) + \mathcal{N}$ according to current policy
        Execute action a_t and observe new state and reward to have $b_t = (s_t, a_t, s_{t+1}, r_t)$ and store it in R
        Sample a mini-batch of N transitions (s_i, a_i, s_{i+1}, r_i)randomly from R
        Set y_i using Equation 3.5
        Update critic by minimizing the loss in (3.4)
        Update the actor policy by (3.7)
        Update $\theta^{Q'}$ and $\theta^{\mu'}$ by (3.6).
    end for
end for

either under attack or normal.Basically, it's built with a deep neural network and approximates the Q-function using temporal-difference learning. The agent takes in state observations from output sensors, and the reward is based on wrong/correct classifications.

For evaluation, they compared their results with datasets from the SWaT and WADI industrial testbeds. The results showed better detection accuracy and fewer false positives than supervised classifiers like SVM and Random Forest. However, their model assumes that the attack types it was trained on will match what it encounters in the real world. This could be a major issue in fast-changing environments where new or unseen attacks.

[76] developed a value-based reinforcement learning algorithm tailored for anomaly detection in CPS. The agent learns a policy that maps system observations to discrete detection decisions, with a reward function explicitly designed to reflect detection accuracy, system stability, and safety-critical constraints. Their approach stands out by integrating domain-specific penalties for false negatives—cases in which attacks are missed—making the model particularly sensitive to stealthy data manipulation. They validated their method on a physical water distribution testbed, showing that the agent could detect integrity attacks more effectively than traditional threshold-based or statistical detectors. Nonetheless, their method heavily depends on a carefully tuned reward structure. Designing such a reward function in more complex or less understood CPS can be non-trivial and

may introduce biases that affect detection sensitivity or lead to reward hacking.

Another RL based anomaly detection was developed in [76], where they used value-functions to learn the optimal policy using SARSA. Therefore, the output is discrete and is considered the detection decision. Their main contribution is in the reward function in which they designed it to reflect detection accuracy and system stability. They penalized the false negatives in order to counter deception attacks. They validated their results on a water distribution testbed, showing that compromising integrity of the data is detectable. However, their method heavily depends on a carefully tuned reward function. Designing such a reward function in different CPS can be challenging.

In [77], they addressed cyberattack resilience with RL within a fixed-time control framework for nonlinear CPS under FDI attacks. Their RL agent acts as a supervisor in a hierarchy control that triggers the backup control policies when certain deviations from nominal behavior are detected. This paper used an on-policy actor-critic agent, where the critic network evaluates system stability, and the actor network generates the control inputs. This framework is built for systems with inaccurate models, in which state estimators alone may fail. However, the detection capability is indirect, relying on deviations rather than direct anomaly detection. So, the system might struggle differentiating between regular behavior, which could lead to more false alarms.

In [78], they implemeneted a detection, identificatoin and mitigation framework using multi-agent deep reinforcement learning (MDRL) for smart grid CPs. They seperated the problem into multiple DRL, each specialized for one of the mentioned 3 tasks. Detection agents use CNN to extract features from sensor data, meanwhile diagnons agent classify the attack types in progress, and response agent choose a recovery party through optimal control configurations. This paper is tested on IEEE bus systems and showed success in achieving the 3 goals. However, training 3 agents is computationally expensive.

In [79], they implemented detection system using the PPO algorithm, which is a policy gradient method with strong training stability. In this case, the detection threshold is a variable that is controlled by the PPO agent, depending on the reward function which consists of detection accuracy, false alarms, and resillience index. They experimented on smart building HVAC systems, which showed that PPO outperformed DQN and A3C to detecting new unseen attacks.

Table 3.1: Comparison of RL-Based Detection Methods for CPS

Reference	RL Method	Application Domain	Attack Type	Key Features
[75]	Double DQN	Industrial CPS	General anomalies	High detection accuracy, benchmarked on real datasets
[76]	Value-based RL	Sensor Networks	Stealthy integrity attacks	Penalizes false negatives, robust to hidden attacks
[77]	Actor-Critic RL	Nonlinear CPS	FDI attacks	Combines detection with resilient control strategies
[78]	Deep RL (multi-layer)	Smart grid CPS	Multiple attack types	Scalable, cascaded detection and recovery layers
[79]	PPO (policy gradient)	Smart grid IDS	Adaptive cyberattacks	Robust policy updates, adapts to evolving threats

3.3 Proposed Dual-Agent Reinforcement Learning Framework

This section introduces a novel hybrid dual-agent architecture that leverages value-based and policy-based RL. The framework is distributed by utilizing the plant side and the command and control (C&C) center, placing a unique agent in each side to ensure full coverage.

3.3.1 System Overview

The proposed framework consists of two cooperative RL agents:

- A DDPG agent located at the plant side, responsible summarizing local system dynamics and build a prediction.
- A DQN agent situated at the command and control center, responsible for interpreting system signals and executing cyber-attacks detection.

This double agent configuration uses local observations at each side and their coordination outputs a global response.

3.3.2 DDPG Agent at the Plant Side

The DDPG agent is embedded at the plant side, where it has access to the actual system outputs. The DDPG agent will have two goals which will be reflected in the action settings, first to predict the system model outputs $y_{predict}$, and to produce an output $p(k) \in [0, 1]$ which is a probability value of an attack. Its observation space includes:

- The actual system output $y(k)$
- The mean value of $y(k)$ and $\tilde{u}(k)$ for a window of size n
- The standard deviation of $y(k)$ and $\tilde{u}(k)$ for a window of size n
- The residual between the predicted values and the actual system values $||y(k) - y_{predict}(k - 1)||$

The final observation vector is defined as

$$[\mu_u(k), \sigma_u(k), \mu_y(k), \sigma_y(k), \|y(k) - \hat{y}(k-1)\|],$$

where μ and σ denote the mean and standard deviation computed over a fixed time window for the input $u(k) \in \mathbb{R}^m$ and output $y(k) \in \mathbb{R}^q$, respectively. The last term captures the absolute prediction error between the current system output and the previously predicted output. Consequently, the total observation dimension is $3q + 2m$.

There are two main objectives to be achieved through designing the reward function which are 1-accurate prediction of the system outputs and 2-maintaining a conservative attack probability estimate. The reward function incorporates a normalized squared error between the true and predicted outputs, along with a regularization term that discourages unnecessarily high attack probabilities, as the DDPG is going to be trained for healthy data first, then FDI attacks on the second stage of training.

Let the true system output at time step k be denoted by the q-dimensional vector $y(k) \in \mathbb{R}^q$, and the predicted output by the DDPG agent be $y_{predict}(k) \in \mathbb{R}^q$.

The normalized squared prediction error is defined as:

$$\mathbf{e}(k) = \left(\frac{\mathbf{y}(k) - \hat{\mathbf{y}}(k)}{y_{\text{max}}} \right)^2. \tag{3.9}$$

To avoid over-penalizing small prediction noise, a tolerance margin ϵ is applied such that errors below this threshold are ignored:

$$\tilde{\mathbf{e}}(k) = \begin{cases} e_i(k)^\alpha & \text{if } e_i(k) > \epsilon^2 \\ 0 & \text{otherwise} \end{cases}, \quad i = 1, 2, ..., q$$

where $\alpha \geq 1$ controls the sensitivity to large errors.

The prediction loss is then computed as the mean of the penalized errors:

$$\mathcal{L}_{\text{pred}}(k) = \frac{1}{2} \sum_{i=1}^{2} \tilde{e}_i(k).$$

In addition, the predicted probability of an attack, denoted $p(k) \in [0, 1]$, is regularized to discourage overly confident attack predictions. The objective is not to enforce $p(k) = 0$, but rather to keep it as small as possible unless strong indicator of an attack exists in the second stage of training. This is achieved using the following quadratic penalty:

$$\mathcal{L}_{\text{attack}}(k) = p(k-1)^2 + r_{dqn} \tag{3.10}$$

where r_{dqn} is the reward from the DQN agent. This is assumed to be available only during training which is an offline training. The final reward function is defined as:

$$r_{ddpg}(k) = -\left(\mathcal{L}_{\text{pred}}(k) + \lambda \cdot \mathcal{L}_{\text{attack}}(k) \right) \tag{3.11}$$

where $\lambda > 0$ is a tunable coefficient that balances prediction accuracy and attack inference.

This reward formulation encourages the agent to produce accurate output predictions while avoiding unwarranted attack alarms, leading to robust anomaly estimation.

3.3.3 DQN Agent at the Command and Control Side

The DQN agent is deployed at the command and control center, where it receives state information and makes detection decisions. Its only goal is to detect any attacks. Its observation space comprises:

- The residual of the corrupted or attacked system output $\tilde{y}(k) - C\hat{X}$
- The original control input $u(k)$, generated by the nominal controller
- The latest action $p(k) + y_p$ generated by the DDPG agent at the plant and might be affected from the attacker by y_p

By incorporating the plant-side signal $p(k)$, the DQN agent gains insight into discrepancies between expected and observed behavior, even in the presence of stealthy attacks. The action space of the DQN agent is simply a discrete detection outputting either a zero or one. The reward system here is simple, no reward $r_{dqn} = 0$ for TPs or TNs, and $r_{dqn} = -1$ for FNs and FPs. Note that FNs and FPs are equally important, since false alarms can lead to unnecessary recovery actions.

This dual-agent architecture is motivated by the following key design principles:

- **Separation of roles:** The DDPG agent focuses on learning continuous dynamics and subtle local patterns, while the DQN agent performs high-level detection based on overall knowledge.
- **Enhanced robustness:** The combined observations from the two sides improve detection robustness against advanced stealthy attacks.

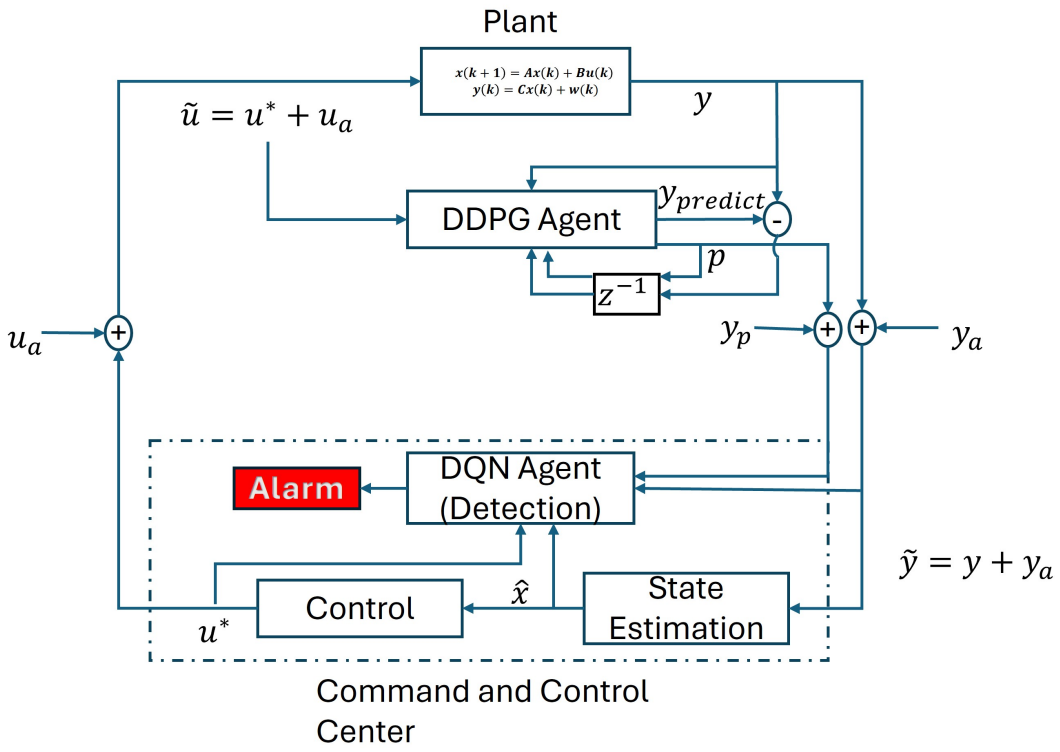

Figure 3.1: Dual-Agent Normal Configuration

3.4 Training of Double Agent and Simulation

The system used for simulation and training is similar to Chapter 2, the quadruple tank from [71]. The quadruple tank was linearized around two equilibrium points, therefore the training will be focused around thouse two points. To improve learning stability and detection robustness, a two-stage training framework is adopted. In **Stage 1**, only the DDPG agent is trained using healthy (attack-free) data. The objective during this phase is twofold: to semi-accurately predict the system output and to generate an auxiliary scalar signal p that remains close to zero under normal operating conditions. During this stage, both the actuator attack signal y_a and the sensor attack signal y_s are set to zero. The reward function is primarily biased toward minimizing the prediction error by assigning a small value to the weighting factor $\lambda = 0.3$, thereby encouraging the agent to focus on model prediction. The training is set to different reference y values, and each episode a reference

value is chosen randomly. To minimize the time of training, a 2-steady state reference values were chosen to train the DDPG on which are the equilibrium points. This would effectively reduce the training time. Practically, for healthy training, practical systems such as power systems have a long history of steady state conditions during peak and normal hours.

In **Stage 2**, During this stage of training, various types of randomized FDI signals were introduced to simulate realistic attack conditions. The injected attack signal $y_a(k)$ was generated using one of several randomly selected strategies at each episode, including: (1) a constant bias drawn from a uniform distribution $y_a(k) = \beta$, where $\beta \sim \mathcal{U}(0, 10)$; (2) a linearly drifting signal $y_a(k) = \beta \cdot k$, with β sampled from $\mathcal{U}(0, 1)$. In addition, a switching strategy was employed to randomly alternate between these attack types across episodes, thereby enhancing the robustness and generalization capability of the agent under diverse FDI scenarios. In this stage, the value of λ is increased to $\lambda = 0.99$, shifting the reward structure to prioritize the transmission of an accurate attack probability to the DQN agent, and ignore the other part of the total reward function. The value of the discount factor γ represents how the agents should take into consideration the future expected reward while updating the value function, and it was set to 0.99 for the DDPG agent, and 0.8 for the DQN agent. This is because collecting data and future predictions need to consider future expected reward, however, detecting the unexpected attacks which might occur at any point in time with no prior knowledge would seem to confuse the DQN agent, hence, the lower value of discount factor.

Table 3.2: Parameter Settings for DDPG and DQN Agents

Parameter	**DDPG Agent**	**DQN Agent**
Observation Dimension	10	9
Action Dimension	3 (continuous)	1 (discrete: 0 or 1)
Observation Space	$\mathbb{R}^{10}$ (Normalized)	$\mathbb{R}^{9}$ (Normalized)
Action Bounds	$[0, 1]^3$	$\{0, 1\}$
Sample Time	1 sec	1 sec
Mini-Batch Size	64	64
Experience Buffer Length	100,000	100,000
Discount Factor γ	0.99	0.8
Target Network Update Frequency	Every 4 steps	Every 4 steps
Policy Update Frequency	Every 10 steps	N/A
Training Episodes	500	
Max Steps Per Episode	2500	
Learning Strategy	Decentralized Multi-Agent Training	

The training results is shown in Figure 3.2.

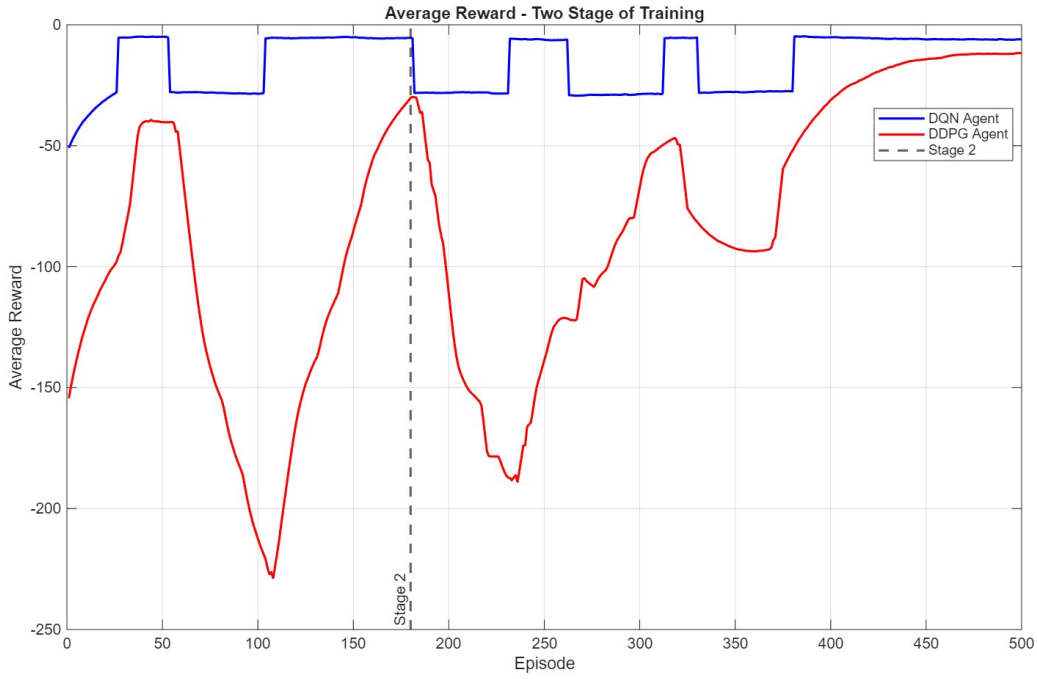

Figure 3.2: Two Stage Training

In Figure 3.2, the first stage of training between $ep = (0, 180)$, the DDPG agent was mainly trained to predict the system outputs. The DDPG average reward kept increasing until it reached a point $(ep = 180)$ which is close as possible to the DQN agent average reward. This is because the main goal is to detect the attacks not to predict the model system, the current prediction model that the agent has will allow him in the future to use it in his future training as a part of the observation to predict an attack probability. In the second stage of the training $ep = (180, 500)$, the DDPG and DQN agent worked together, in which both of them performed poorly $ep = (180, 225)$, but then the average reward kept increasing which means the performance was improving constant untill it reached at $ep = 380$, in which both of the agents seemed to be able to detect the attacks together, since the rewards converged to a high value. The main goal of the DDPG agent is to help the DQN agent detect any cyber attacks in the future. This is done in the second stage, as the training figure shows that both are converging together to an optimal policy.

3.5 Simulation Detection Performance Metrics and Other Methods Comparisons

To evaluate detection performance, the agents were tested over 200 additional episodes using unseen data. A mix of healthy and FDI attacks scenarios was introduced randomly against sensors or actuators or both. In the following section, we explain how the random attacks are generated

3.5.1 Randomized Attacks Generation

Let the system have $n_s = 2$ sensors and $n_a = 2$ actuators and denote the set of all channels by

$$\mathcal{C} = \{1, \ldots, N\}, \qquad N = n_s + n_a = 4.$$

The randomized attack for each experiment is generated as follows.

1. Number of attacked channels Select the number of attacked channels K as a discrete uniform random variable

$$K \sim \mathcal{U}\{1, 2, \ldots, 4\}.$$

If $K = 4$ then all channels are attacked; if $K < 4$ then exactly K distinct channels are chosen to be attacked.

2. Channel selection Given $K = k$, choose the attack set $S \subseteq \mathcal{C}$ uniformly at random among all $\binom{N}{k}$ subsets of cardinality k:

$$\mathbb{P}(S = s \mid K = k) = \frac{1}{\binom{N}{k}}, \qquad s \subseteq \mathcal{C},\ |s| = k.$$

3. Attack type and magnitude For each selected channel $i \in S$ independently:

- pick the attack type $\tau_i \in \{\text{step}, \text{ramp}\}$ (e.g. $\mathbb{P}(\tau_i = \text{step}) = \mathbb{P}(\tau_i = \text{ramp}) = 0.5$),

- draw the attack amplitude A_i from a prescribed distribution, e.g.

$$A_i \sim \mathcal{U}[0.001, 1],$$

- The attack start time, and duration are fixed to $t = (700, 1000)$ since it has no benefit to randomize it.

The FDI attacks introduced are either a step input or a ramp input, chosen randomly at equal probability. The ramp attack signal

$$P(t) = A(t - 700)$$

and the

$$P(t) = A$$

can be targeted at any channel depending on the previous random generation. An example of an FDI attack aiming the first sensor only introduced in testing Episode 13:

$$y_a(t) = \begin{cases} [0, 0]^T, & 0 \leq t < 700, \\ [-0.005(t - 700), 0]^T, & 700 \leq t < 1000, \\ [0, 0]^T, & t \geq 1000. \end{cases}$$

3.5.2 Results of Double Agent Detection Against FDI Attacks

The key performance indicators of the detector against the test FDI attacks and Chapter 2 data-driven covert attack are summarized in Table 3.3.

Table 3.3: Evaluation Metrics for Attack Detection Against FDI Attack and Chapter 2 Data-Driven Covert Attack

Metric	**FDI**	**Data-Driven Covert Attack($m = 0.01$) - Chapter 2**
True Positive Rate (Detection Recall)	92.5%	60.5%
False Positive Rate	6.2%	6.1%
True Negative Rate	93.8%	93.9%
False Negative Rate	7.5%	39.5%
F1-Score	0.928	0.63
Accuracy	93.2%	80.5%

The results indicate that the DQN agent, informed by the DDPG-generated scalar signal $p(k)$, can effectively distinguish between normal and compromised behavior. The low false alarm rate (6.2%) and high true positive rate (92.5%) demonstrate the robustness of the proposed architecture against FDI attacks. However, for covert attacks, the scores are a little bit lower which is acceptable because they are originally smart attacks, any detection of these attacks are considered acceptable. Note that if the true positive rate above 50%, the attack is considered as detected because it means most of the attack duration the detector is generating an alarm.

3.5.3 Comparison with Other Detection Methods

For benchmarking, the proposed method was compared with three approaches against FDI attacks, Replay attacks and Covert attacks:

(1) A chi-square residual detector using a Kalman filter.

(2) A standard watermarking detection with chi-square resulting in 15% performance degradation.

(3) A benchmark from double Q-network detection [75]

Table 3.4: Comparison of Detection Accuracy against FDI Attacks

Detection Method	**Detection Rate (%)**	**False Alarm Rate (%)**	**F1-Score (%)**
Chi-Square	96.2	5.1	86.4
Watermarking	96.4	5.1	85.8
Double Q-Network [75]	97.1	2.1	89.4
Proposed Dual-Agent RL	**94.2**	**3.3**	**84.1**

Table 3.5: Comparison of Detection Accuracy against Replay Attacks

Detection Method	**Detection Rate (%)**	**False Alarm Rate (%)**	**F1-Score (%)**
Chi Square	8.92	5.2	14.45
Watermarking	79	5.1	78
Double Q-Network [75]	8	2.4	12.2
Proposed Dual-Agent RL	**82.1**	**3.44**	**77.1**

Table 3.6: Comparison of Detection Accuracy against Data-Driven Covert Attack(m=0.01)

Detection Method	**Detection Rate (%)**	**False Alarm Rate (%)**	**F1-Score (%)**
Chi Square	8.94	5.1	14.6
Watermarking	8.8	4.9	13.3
Double Q-Network [75]	5.1	1.99	8.26
Proposed Dual-Agent RL	**78.2**	**3.56**	**74.4**

The comparative performance of different detection algorithms against three representative classes of cyber-attacks—namely False Data Injection (FDI), Replay, and Data-Driven Covert Attacks—is summarized in Tables 3.4–3.6. The evaluation focuses on three performance indicators: Detection Rate (DR), False Alarm Rate (FAR), and the harmonic mean of precision and recall represented by the F1-Score. These metrics jointly reflect the accuracy, reliability, and robustness of each detection strategy under various adversarial conditions.

1. False Data Injection (FDI) Attacks. From Table 3.4, the classical Chi-Square and watermarking-based detectors achieve detection rates of approximately 96%, with moderate false alarm rates near 5%. The Double Q-Network [75] slightly improves the detection rate to 97.1% and achieves the lowest FAR of 2.1%, resulting in the highest F1-score (89.4%) among traditional learning-based baselines. In contrast, the Proposed Dual-Agent Reinforcement Learning (RL) method exhibits a detection rate of 94.2% and a FAR of 3.3%, leading to an F1-score of 84.1%. Although its performance against FDI attacks is slightly lower than the single-agent DQN-based detector, this outcome is expected. The dual-agent framework is not solely optimized for classical FDI attacks, but rather for generalizable detection across diverse and stealthy data-driven scenarios. Thus, the minor performance reduction under FDI conditions reflects a trade-off favoring adaptability rather than overfitting to a specific attack type.

2. Replay Attacks. The results under replay attacks, presented in Table 3.5, highlight the superior adaptability of the proposed method. The Chi-Square and Double Q-Network detectors demonstrate limited capability in detecting replayed signals, with detection rates below 10%, indicating their vulnerability to temporally correlated attack sequences that mimic legitimate system dynamics. Watermarking significantly improves replay detection, achieving a detection rate of 79% and an F1-score of 78%. However, the Proposed Dual-Agent RL model surpasses all baselines with a detection rate of 82.1% and the lowest false alarm rate (3.44%), yielding an overall F1-score of 77.1%. This improvement illustrates the model's ability to capture temporal dependencies and nonlinear relationships between sensor and actuator behaviors, which are critical for identifying replayed or time-shifted patterns that evade statistical detectors.

3. Data-Driven Covert Attacks. The most significant performance contrast is observed under the data-driven covert attack scenario (Chapter 2), as shown in Table 3.6. These attacks are inherently stealthier, designed to manipulate sensor and actuator data while maintaining statistical consistency with nominal behavior, thereby deceiving conventional detectors. Both the Chi-Square and watermarking methods perform poorly, with detection rates below 9% and F1-scores around 14%. Even the Double Q-Network baseline exhibits very limited detection capability (5.1% DR),

suggesting that single-agent learning models struggle to generalize under covert and adaptive adversarial strategies. In contrast, the Proposed Dual-Agent RL detector achieves a detection rate of 78.2% and an F1-score of 74.4%, while maintaining a low FAR of 3.56%. These results demonstrate a substantial improvement of more than 60 percentage points in detection accuracy over one of the data-driven detection methods. The dual-agent structure—combining a policy-learning (actor) and evaluation (critic) mechanism—enables continuous adaptation and cooperation between control and observation spaces, allowing the system to identify subtle inconsistencies introduced by covert attacks.

4. Cross-Attack Generalization. Analyzing the results across all three attack types, it becomes evident that traditional statistical detectors such as the Chi-Square test are effective primarily for large-magnitude, uncorrelated FDI-type disturbances but fail to identify temporally consistent or data-driven attacks. Watermarking extends the detection scope by introducing random excitation to the control input, enhancing replay detection but remaining limited when the attacker adapts to the covert attack. On the other hand, deep reinforcement learning–based approaches, particularly the proposed dual-agent model, demonstrate remarkable adaptability. While its performance is slightly lower than DQN under simple FDI attacks, it vastly outperforms all baselines under replay and covert scenarios, indicating that the proposed architecture learns dynamic representations of system behavior rather than static residual thresholds.

5. Summary of Findings. Overall, the proposed Dual-Agent RL framework provides a significant advancement in the detection of stealthy cyber-attacks on cyber-physical systems. The model's strong generalization across attack types confirms its robustness, and the low false alarm rate indicates stable learning and minimal overfitting. These findings validate the effectiveness of integrating dual-agent reinforcement learning into security-aware control frameworks, where both the observation and control layers cooperate to preserve system integrity against unknown or data-driven adversaries.

6. Key Observations.

- The proposed method achieves over **78%** detection accuracy against covert attacks, compared to less than **10%** for conventional detectors.
- The false alarm rate consistently remains below **3.6%**, ensuring high reliability and minimal disruption to normal operation.
- The method's performance remains stable across attack types, confirming strong cross-domain generalization.

In conclusion, the Dual-Agent RL model demonstrates not only superior adaptability and robustness against sophisticated stealthy attacks but also provides a foundation for future research on cooperative multi-agent learning for cyber-physical system security.

The dual-agent approach consistently outperformed traditional detection strategies, especially in scenarios involving novel or composite FDI attacks. Its capability to generalize beyond seen data stems from the use of environment interaction, rather than static training sets.

Chapter Summary

This chapter presented a novel dual-agent detection architecture that combines the complementary strengths of policy-based and value-based reinforcement learning to enhance CPS security. Intelligence is distributed across two strategic locations: a DDPG agent located at the plant side for local dynamics estimation and anomaly summarization, and a DQN agent at the (C&C) center for centralized detection decisions. The separation of roles allows for fine-grained, locally informed observations while preserving a global decision-making capability, thereby improving robustness against sophisticated and stealthy attacks.

The DDPG agent at the plant side is designed to learn continuous system dynamics and to produce two outputs: a multi-dimensional prediction of the system outputs y_{predict} and a scalar probability-like indicator $p(k) \in [0, 1]$ that encodes the agent's local belief in the presence of an attack. Its observation vector aggregates short-term statistics (means and standard deviations) of inputs and outputs over a sliding window together with the prediction residual $\|y(k) - \hat{y}(k-1)\|$. A dedicated reward function balances prediction accuracy against false-positive avoidance by

combining a normalized, thresholded squared prediction error with a regularization term on the attack-probability output; the tunable coefficient λ controls this trade-off.

The DQN agent at the C&C center consumes the potentially corrupted measurements $\tilde{y}(k)$, the nominal controller input $u(k)$, and the plant-side signal $p(k)$ to make binary detection decisions. By incorporating the plant-side summary $p(k)$, the DQN agent gains an informative, localized cue that improves its ability to discriminate between nominal disturbances and cyber attacks. The DQN's reward structure is deliberately sparse and conservative, penalizing false negatives and false positives to encourage reliable detection behavior.

A two-stage training regimen was used to stabilize learning and promote generalization. In Stage 1, the DDPG agent is trained on healthy data to learn a predictive model and to produce a near-zero $p(k)$ under nominal conditions (small λ emphasis on prediction). In Stage 2, randomized FDI scenarios are introduced and λ is increased so that the DDPG's auxiliary signal becomes informative for detection; concurrently the DQN is trained to exploit discrepancies between predicted and observed behavior. This curriculum—first modeling, then adversarial adaptation—improves detection robustness while mitigating catastrophic forgetting.

Simulation results on the quadruple-tank benchmark demonstrate that the dual-agent framework is able to learn cooperative policies: the DDPG attains accurate local predictions and produces meaningful local indicators, while the DQN integrates these signals to raise reliable alarms. Training curves (Figure 3.2) show initial model-learning convergence followed by a coordinated increase in detection performance during adversarial training. Parameter settings and architecture dimensions used in the experiments are summarized in Table 3.2.

In summary, the chapter establishes the dual-agent architecture as an effective paradigm for CPS anomaly detection: the DDPG provides localized dynamical insight and conservative attack scoring, and the DQN fuses these cues with global measurements to make robust detection decisions. The proposed two-stage training strategy yields stable learning and improved generalization to unseen cyber attacks and untrained for.

Chapter 4

Conclusion and Future Directions

This thesis dives into the growing vulnerability of CPS to complex cyber-attacks and introduces a fresh attack strategy along with a detection mechanism to counter it. You will notice two key take-aways from this work. Firstly, we developed a data-driven attack that is able to bypass all traditional detectors, and it doesn't even rely on complicated system dynamics or estimation algorithms, where no mondel estimation is needed or model knowledge requirement.

The proposed DDCRA exploits input-output recordings during normal operation to construct the attack signal, this attack signal will basically a scaled form of the previous recorded signals. The attack signal on the output channel will be a scaled recording of previous output channels, and the input is similarly a scaled previous recording of control inputs that have generated the same recorded outputs. This attack bypasses traditional residual-based methods, including chi-square detectors, and watermarking. Simulations on a quadruple-tank benchmark system confirmed the stealthiness and effectiveness of the DDCRA including scenarios where gaussian noise is presented.

To address the limitations of model-based detection which was shown in the second chapter, this thesis introduced a novel dual-agent RL framework. The DDPG agent locally observes actual output and the potentially altered input, generating a system prediction and anomaly score that is transmitted to the DQN agent at the command and control side. The DQN agent makes the final detection decision after receiving the feedback from the DDPG agent and evaluating the residual values. This dual agent architecture allows for an enhanced overview of the environment resulting

in better decision making process. Simulation results demonstrate that the RL-based method significantly improves detection rates in the presence of covert, replay and FDI attacks, outperforming traditional detection approaches.

Future Work

The research presented in this thesis opens several promising directions for future investigation. In Chapter 2, the proposed data-driven covert attack was primarily designed to demonstrate that a stealthy and effective covert attack can be executed without any prior model knowledge of the target system. However, the mechanism of this attack—particularly the adaptive excitation of system inputs—can also be exploited for enhanced system identification. Since the injected excitation signal in the covert attack is stronger and more informative than typical identification signals, yet remains undetectable by standard monitoring schemes, this framework could potentially enable faster or more accurate estimation of system dynamics. Although model discovery was beyond the scope of the present work, this alternative direction represents a compelling opportunity for extending the concept of data-driven covert attacks toward model learning applications and how to defend against it.

Moreover, the attack framework developed in Chapter 2 could serve as a foundation for evaluating and challenging existing covert-attack defense strategies that rely heavily on dynamic model variation, such as Moving Target Defense (MTD). In such defenses, it is often assumed that the attacker lacks knowledge of the auxiliary or time-varying models and has insufficient time to estimate them. However, a data-driven covert attacker capable of recording sufficient data across multiple model configurations—and equipped with partial knowledge of the defense mechanism—may be able to construct a more sophisticated, adaptive attack. Exploring this line of work could provide valuable insights into the robustness limits of model-based defensive architectures and inform the design of more resilient CPS protection mechanisms.

In Chapter 3, the proposed Dual-Agent Reinforcement Learning detector demonstrated strong capabilities in detecting various types of model-based and data-driven cyber-attacks. A natural extension of this framework would involve expanding it beyond detection to include fault and attack

isolation—specifically, identifying which sensor or actuator channel is under attack. To achieve this, additional input features could be introduced to the DDPG agent on the plant side, such as residuals derived from unknown input observers (UIOs) or other state estimation mechanisms. These auxiliary signals would enrich the agent's perception of the system state and could provide the DQN agent with higher-level diagnostic information. Furthermore, the neural architecture could be extended to include multiple output heads, one dedicated to attack detection and another to attack localization or classification. Implementing and training such an enhanced multi-task reinforcement learning framework would require substantial additional time and computational resources, but it represents an exciting and meaningful future research direction for developing comprehensive autonomous CPS defense systems.

Bibliography

[1] A. K. Tyagi and N. Sreenath, "Cyber physical systems: Analyses, challenges and possible solutions," *Internet of Things and Cyber-Physical Systems*, vol. 1, pp. 22–33, 2021.

[2] P. Leitão, A. W. Colombo, and S. Karnouskos, "Industrial automation based on cyber-physical systems technologies: Prototype implementations and challenges," *Computers in Industry*, vol. 81, pp. 11–25, 2016.

[3] E. D. Sontag, *Mathematical Control Theory: Deterministic Finite Dimensional Systems*. Berlin, Germany: Springer, 2013, vol. 6.

[4] J. H. Davis, "Luenberger observers," in *Foundations of Deterministic and Stochastic Control*, 2002, pp. 245–254.

[5] M. S. Grewal and A. P. Andrews, *Kalman Filtering: Theory and Practice Using MATLAB*, 4th ed. John Wiley & Sons, 2014.

[6] K. Fujii, "Extended kalman filter," Joint Linear Collider Workshop (JLC), Tech. Rep., 2013, accessed: Mar. 4, 2026. [Online]. Available: https://www-jlc.kek.jp/2004sep/subg/offl/kaltest/doc/ReferenceManual.pdf

[7] J. Zabczyk, *Mathematical Control Theory*. Springer, 2020.

[8] B. S. Anjali, A. Vivek, and J. L. Nandagopal, "Simulation and analysis of integral lqr controller for inner control loop design of a fixed wing micro aerial vehicle (mav)," *Procedia Technology*, vol. 25, pp. 76–83, 2016.

[9] R. Rajkumar, I. Lee, L. Sha, and J. Stankovic, "Cyber-physical systems: The next computing revolution," in *Proceedings of the 47th Design Automation Conference*, 2010, pp. 731–736.

[10] K. Pannerselvam and S. Rajiakodi, "Towards smarter, interconnected futures: The crucial role of data in cyber-physical systems," in *Intelligent Cyber-Physical Systems for Healthcare Solutions*. Springer, 2024, pp. 181–194.

[11] 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.

[12] 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.

[13] B. Milosevic, C. Fischione, and K. H. Johansson, "Analysis and mitigation of bias injection attacks against kalman filter," *IEEE Transactions on Control of Network Systems*, vol. 5, no. 3, pp. 1075–1085, 2018.

[14] H. Lou, B. Jiang, Z. Wu, S. Fan, and X. Li, "Performance-based event-triggered observer design for security control of cyber-physical systems with malice attacks through adaptive sliding mode approach," *Discrete and Continuous Dynamical Systems - Series S*, vol. 17, no. 9, pp. 2855–2871, 2024.

[15] N. Zhang, W. Qi, G. Pang, J. Cheng, and K. Shi, "Observer-based sliding mode control for fuzzy stochastic switching systems with deception attacks," *Applied Mathematics and Computation*, vol. 427, p. 127153, 2022.

[16] M. Tranninger, H. Niederwieser, R. Seeber, and M. Horn, "Unknown input observer design for linear time-invariant systems—a unifying framework," *International Journal of Robust and Nonlinear Control*, vol. 33, no. 15, pp. 8911–8934, 2023.

[17] 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.

[18] N. Tomasevic, N. Gvozdenovic, and S. Vranes, "An overview and comparison of supervised data mining techniques for student exam performance prediction," *Computers & Education*, vol. 143, p. 103676, 2020.

[19] M. S. Amin, Y. K. Chiam, and K. D. Varathan, "Identification of significant features and data mining techniques in predicting heart disease," *Telematics and Informatics*, vol. 36, pp. 82–93, 2019.

[20] A. Ishaq, S. Sadiq, M. Umer, S. Ullah, S. Mirjalili, V. Rupapara, and M. Nappi, "Improving the prediction of heart failure patients' survival using smote and effective data mining techniques," *IEEE Access*, vol. 9, pp. 39 707–39 716, 2021.

[21] D. van den Bergh, M. A. Clyde, A. R. K. N. Gupta, T. de Jong, Q. F. Gronau, M. Marsman, A. Ly, and E.-J. Wagenmakers, "A tutorial on bayesian multi-model linear regression with bas and jasp," *Behavior Research Methods*, pp. 1–21, 2021.

[22] Y. Chen, W. Zheng, W. Li, and Y. Huang, "Large group activity security risk assessment and risk early warning based on random forest algorithm," *Pattern Recognition Letters*, vol. 144, pp. 1–5, 2021.

[23] J. Yan, B. Tang, and H. He, "Detection of false data attacks in smart grid with supervised learning," in *2016 International Joint Conference on Neural Networks (IJCNN)*, 2016, pp. 1395–1402.

[24] Y. Dong, X. Ma, and T. Fu, "Electrical load forecasting: A deep learning approach based on k-nearest neighbors," *Applied Soft Computing*, vol. 99, p. 106900, 2021.

[25] J. S. Manoharan, "Study of variants of extreme learning machine (elm) brands and its performance measure on classification algorithm," *Journal of Soft Computing Paradigm (JSCP)*, vol. 3, no. 2, pp. 83–95, 2021.

[26] W. Pu, "Shuffle gan with autoencoder: A deep learning approach to separate moving and stationary targets in sar imagery," *IEEE Transactions on Neural Networks and Learning Systems*, vol. 33, no. 9, pp. 4770–4784, 2021.

[27] H. Wang, Y. Shao, S. Zhou, C. Zhang, and N. Xiu, "Support vector machine classifier via $l_{0/1}$ soft-margin loss," *IEEE Transactions on Pattern Analysis and Machine Intelligence*, vol. 44, no. 10, pp. 7253–7265, 2021.

[28] B. Amrutha, I. Meghana, R. Tejas, H. V. Pilare, and D. Annapurna, "An efficient automated intrusion detection system using hybrid decision tree," in *Inventive Systems and Control: Proceedings of ICISC 2022*, 2022, pp. 703–716.

[29] A. K. Onaolapo, R. P. Carpanen, D. G. Dorrell, and E. E. Ojo, "A comparative assessment of conventional and artificial neural networks methods for electricity outage forecasting," *Energies*, vol. 15, no. 2, p. 511, 2022.

[30] ——, "Event-driven power outage prediction using collaborative neural networks," *IEEE Transactions on Industrial Informatics*, vol. 19, no. 3, pp. 3079–3087, 2022.

[31] A. Onaolapo, R. P. Carpanen, D. Dorrell, and E. Ojo, "Forecasting electricity outage in kwazulu-natal, south africa using trend projection and artificial neural networks techniques," in *2021 IEEE PES/IAS PowerAfrica*, 2021, pp. 1–5.

[32] A. K. Onaolapo, R. P. Carpanen, D. G. Dorrell, and E. E. Ojo, "Transmission line fault classification and location using multi-layer perceptron artificial neural network," in *IECON 2020: The 46th Annual Conference of the IEEE Industrial Electronics Society*, 2020, pp. 5182–5187.

[33] A. Onaolapo, R. Pillay-Carpanen, D. Dorrell, and E. Ojo, "A comparative evaluation of conventional and computational intelligence techniques for forecasting electricity outage," in *2021 Southern African Universities Power Engineering Conference/Robotics and Mechatronics/Pattern Recognition Association of South Africa (SAUPEC/RobMech/PRASA)*, 2021, pp. 1–6.

[34] D. Sarvamangala and R. V. Kulkarni, "Convolutional neural networks in medical image understanding: A survey," *Evolutionary Intelligence*, vol. 15, no. 1, pp. 1–22, 2022.

[35] C. L. Srinidhi, O. Ciga, and A. L. Martel, "Deep neural network models for computational histopathology: A survey," *Medical Image Analysis*, vol. 67, p. 101813, 2021.

[36] 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.

[37] D. C. Le, N. Zincir-Heywood, and M. Heywood, "Training regime influences to semi-supervised learning for insider threat detection," in *2021 IEEE Security and Privacy Workshops (SPW)*, 2021, pp. 13–18.

[38] 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)*, 2021, pp. 1–6.

[39] H. Moudoud, Z. Mlika, L. Khoukhi, and S. Cherkaoui, "Detection and prediction of fdi attacks in iot systems via hidden markov model," *IEEE Transactions on Network Science and Engineering*, vol. 9, no. 5, pp. 2978–2990, 2022.

[40] D. Nguyen, R. Vadaine, G. Hajduch, R. Garello, and R. Fablet, "Geotracknet—a maritime anomaly detector using probabilistic neural network representation of ais tracks and a contrario detection," *IEEE Transactions on Intelligent Transportation Systems*, vol. 23, no. 6, pp. 5655–5667, 2022.

[41] H. Zhao, J. Liu, H. Chen, J. Chen, Y. Li, J. Xu, and W. Deng, "Intelligent diagnosis using continuous wavelet transform and gauss convolutional deep belief network," *IEEE Transactions on Reliability*, vol. 72, no. 2, pp. 692–702, 2023.

[42] S. Ahmed, Y. Lee, S.-H. Hyun, and I. Koo, "Unsupervised machine learning-based detection of covert data integrity assault in smart grid networks utilizing isolation forest," *IEEE Transactions on Information Forensics and Security*, vol. 14, no. 10, pp. 2765–2777, 2019.

[43] Z. Qu, X. Bo, T. Yu, Y. Liu, Y. Dong, Z. Kan, L. Wang, and Y. Li, "Active and passive hybrid detection method for power cps false data injection attacks with improved akf and gru-cnn," *IET Renewable Power Generation*, vol. 16, no. 7, pp. 1490–1508, 2022.

[44] T. Vaiyapuri, H. Aldosari, G. Alharbi, Y. Bouteraa, G. P. Joshi, and W. Cho, "Metaheuristics based dimensionality reduction with deep learning driven false data injection attack detection for enhanced network security," *Scientific Reports*, vol. 14, no. 1, p. 18967, 2024.

[45] S. Padhan and A. K. Turuk, "Design of false data injection attacks in cyber-physical systems," *Information Sciences*, vol. 608, pp. 825–843, 2022.

[46] W. Tu, J. Dong, and D. Zhai, "Optimal epsilon-stealthy attack in cyber-physical systems," *Journal of the Franklin Institute*, vol. 358, no. 1, pp. 151–171, 2021.

[47] T.-Y. Zhang and D. Ye, "False data injection attacks with complete stealthiness in cyber-physical systems: A self-generated approach," *Automatica*, vol. 120, p. 109117, 2020.

[48] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," in *Preprints of the 1st Workshop on Secure Control Systems*, 2010.

[49] A. Teixeira, I. Shames, H. Sandberg, and K. H. Johansson, "Revealing stealthy attacks in control systems," in *2012 50th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2012, pp. 1806–1813.

[50] N. Falliere, L. O. Murchu, E. Chien *et al.*, "W32. stuxnet dossier," *White paper, symantec corp., security response*, vol. 5, no. 6, p. 29, 2011.

[51] R. Langner, "Stuxnet: Dissecting a cyberwarfare weapon," *IEEE Security & Privacy*, vol. 9, no. 3, pp. 49–51, 2011.

[52] Y. Mo, S. Weerakkody, and B. Sinopoli, "Physical authentication of control systems: Designing watermarked control inputs to detect counterfeit sensor outputs," *IEEE Control Systems Magazine*, vol. 35, no. 1, pp. 93–109, 2015.

[53] B. Satchidanandan and P. R. Kumar, "Dynamic watermarking: Active defense of networked cyber-physical systems," *Proceedings of the IEEE*, vol. 105, no. 2, pp. 219–240, 2016.

[54] M. Zhou, Z. Zhang, and L. Xie, "Permutation entropy based detection scheme of replay attacks in industrial cyber-physical systems," *Journal of the Franklin Institute*, vol. 358, no. 7, pp. 4058–4076, 2021.

[55] S. Gargoum, N. Yassaie, A. W. Al-Dabbagh, and C. Feng, "A data-driven framework for verified detection of replay attacks on industrial control systems," *IEEE Transactions on Automation Science and Engineering*, 2024.

[56] S. Weerakkody and B. Sinopoli, "Detecting integrity attacks on control systems using a moving target approach," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 5820–5826.

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

[58] A. Eslami and K. Khorasani, "Cyber-attack detection and isolation in event-based cyber-physical systems," in *2024 American Control Conference (ACC)*, 2024, pp. 1–7.

[59] J. Kim and H. Shim, "A countermeasure against zero-dynamics sensor attack via generalized hold feedback," in *2019 58th Annual Conference of the Society of Instrument and Control Engineers of Japan (SICE)*, 2019, pp. 663–668.

[60] Z. Zhang, H. Li, and Y. Todo, "Zero-dynamics attack detection based on data association in feedback pathway," *Cognitive Robotics*, vol. 5, pp. 126–139, 2025.

[61] J. Gao, Y. Wang, Z. Zuo, and W. Zhang, "Zero-dynamics attacks for multi-agent systems without velocity measurements," *Journal of Systems Science and Complexity*, vol. 37, no. 5, pp. 1809–1831, 2024.

[62] M. Ghaderi, K. Gheitasi, and W. Lucia, "A novel control architecture for the detection of false data injection attacks in networked control systems," in *2019 American Control Conference (ACC)*. IEEE, 2019, pp. 139–144.

[63] D. Mikhaylenko and P. Zhang, "Stealthy local covert attacks on cyber-physical systems," *IEEE Transactions on Automatic Control*, vol. 67, no. 12, pp. 6778–6785, 2021.

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

[65] A. Barboni, H. Rezaee, F. Boem, and T. Parisini, "Distributed detection of covert attacks for interconnected systems," in *2019 18th European Control Conference (ECC)*. IEEE, 2019, pp. 2240–2245.

[66] D. I. Urbina, J. A. Giraldo, A. A. Cardenas, N. O. Tippenhauer, J. Valente, M. Faisal, J. Ruths, R. Candell, and H. Sandberg, "Limiting the impact of stealthy attacks on industrial control systems," in *Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security*, 2016, pp. 1092–1105.

[67] G. Dán and H. Sandberg, "Stealth attacks and protection schemes for state estimators in power systems," in *2010 First IEEE International Conference on Smart Grid Communications*. IEEE, 2010, pp. 214–219.

[68] A. A. Taha and A. Hanbury, "Metrics for evaluating 3d medical image segmentation: analysis, selection, and tool," *BMC Medical Imaging*, vol. 15, pp. 1–28, 2015.

[69] H. Zhang and A. Teixeira, "Stealthy targeted local covert attacks on cyber-physical systems," *Automatica*, vol. 157, p. 111104, 2024.

[70] X.-X. Ren, G.-H. Yang, and X.-G. Zhang, "Optimal stealthy attack with historical data on cyber-physical systems," *Automatica*, vol. 151, p. 110895, 2023.

[71] K. H. Johansson, "The quadruple-tank process: A multivariable laboratory process with an adjustable zero," *IEEE Transactions on Control Systems Technology*, vol. 8, no. 3, pp. 456–465, 2002.

[72] Y. Mo and B. Sinopoli, "Secure control against replay attacks," in *2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton)*, 2009, pp. 911–918.

[73] V. Mnih, K. Kavukcuoglu, D. Silver, A. A. Rusu, J. Veness, M. G. Bellemare, A. Graves, M. Riedmiller, A. K. Fidjeland, G. Ostrovski *et al.*, "Human-level control through deep reinforcement learning," *Nature*, vol. 518, no. 7540, pp. 529–533, 2015.

[74] T. P. Lillicrap, J. J. Hunt, A. Pritzel, N. Heess, T. Erez, Y. Tassa, D. Silver, and D. Wierstra,

"Continuous control with deep reinforcement learning," *arXiv preprint arXiv:1509.02971*, 2015.

[75] Y. Zhang, M. Jamjoom, and Z. Ullah, "Double deep q-network next-generation cyber-physical systems: A reinforcement learning-enabled anomaly detection framework for next-generation cyber-physical systems," *Electronics*, vol. 12, no. 17, p. 3632, 2023.

[76] M. Ibrahim and R. Elhafiz, "Security analysis of cyber-physical systems using reinforcement learning," *Sensors*, vol. 23, no. 3, p. 1634, 2023.

[77] M. Mazare, "Reinforcement learning-based fixed-time resilient control of nonlinear cyber physical systems under false data injection attacks and mismatch disturbances," *Journal of the Franklin Institute*, vol. 360, no. 18, pp. 14 926–14 938, 2023.

[78] K. Zhong, Z. Yang, S. Yu, and K. Li, "Deep reinforcement learning-based multi-layer cascaded resilient recovery for cyber-physical systems," *IEEE Transactions on Services Computing*, 2024.

[79] M. Ishaque, M. G. M. Johar, A. Khatibi, and M. Yamin, "Dynamic adaptive intrusion detection system using hybrid reinforcement learning," in *International Conference on Business and Technology*. Springer, 2023, pp. 245–253.