Computers and Chemical Engineering 180 (2024) 108498

Contents lists available at ScienceDirect

Computers and Chemical Engineering

journal homepage: www.elsevier.com/locate/cace

Integrating machine learning detection and encrypted control for enhanced cybersecurity of nonlinear processes

Yash A. Kadakia [a], Atharva Suryavanshi [a], Aisha Alnajdi [b], Fahim Abdullah [a], Panagiotis D. Christofides [a,b,*]

[a] Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA, 90095-1592, USA
[b] Department of Electrical and Computer Engineering, University of California, Los Angeles, CA 90095-1592, USA

ARTICLE INFO

Keywords:
Model predictive control
Encrypted control
Machine learning
Cybersecurity
Semi-homomorphic encryption
Quantization

ABSTRACT

This study presents an encrypted two-tier control architecture integrated with a machine learning (ML) based cyberattack detector to enhance the operational safety, cyber-security, and performance of nonlinear processes. The upper tier of this architecture employs an encrypted nonlinear Lyapunov-based model predictive controller (LMPC) to enhance closed-loop performance, while the lower tier utilizes an encrypted set of linear controllers to stabilize the process. Encrypted signals from the sensors are decrypted at the upper tier for plain text control input computation, while the lower tier computes control inputs in an encrypted space, due to its exclusive use of linear operations. While this design enhances closed-loop performance, it exposes the upper tier to potential cyberattacks. To mitigate this risk, an ML-based detector is developed in the form of a feed-forward neural network, utilizing sensor-derived data for attack detection. Upon attack detection, the control system logic deactivates the performance-enhancing upper tier and relies solely on the cybersecure lower tier for system stabilization. The study also includes a comprehensive stability analysis of the two-tier control structure, establishing error bounds related to quantization and sample-and-hold controller implementations. The proposed control framework can be extended to any nonlinear process that is controlled by a combination of linear and nonlinear controllers to enhance the system cybersecurity. Guidelines such as quantization parameter selection, cyberattack detector development, and sampling time criteria are included to facilitate practical implementation. Simulation results of a nonlinear chemical process network demonstrated the robustness of the encrypted control architecture and cyberattack detector, as well as its ability to detect previously unseen attack patterns.

1. Introduction

The swift advancements in technology and the increasing integration of devices have made interconnected cyber–physical systems essential elements of vital infrastructure in various sectors like energy, water, transportation, and manufacturing. In particular, systems that employ SCADA (Supervisory Control and Data Acquisition) technology play a crucial role in overseeing, directing, and automating intricate operations, thereby boosting efficiency and productivity. Nonetheless, the expanded interlinking and fusion of SCADA systems with the internet and corporate networks have made them susceptible to potential cyber threats. A breach or compromise within these systems could lead to grave outcomes, including disruption of essential services, physical harm, financial setbacks, and even jeopardizing public safety. Current advancements in cyberattack methodologies underscore the importance of instituting robust cybersecurity measures.

While notable strides have been made in tackling cybersecurity issues within the domain of information technology (IT), the operational technology (OT) domain is currently lagging behind in terms of advancements. IT predominantly concentrates on the software aspect of systems, covering areas like network architecture and data administration. On the other hand, OT is responsible for maintaining the seamless functioning of essential infrastructure, such as power grids, intelligent meters, and distribution networks. Cyberattacks on OT infrastructure can result in consequences such as operational shutdowns, service disruptions, data leaks, and potentially catastrophic explosions. As an illustration, consider the case of the Stuxnet malware, which was uncovered in 2010. This particular malicious software was designed with a specific focus on infiltrating SCADA systems. Stuxnet managed to breach programmable logic controllers (PLCs) within Iranian nuclear facilities, collecting valuable information about the industrial system

* Corresponding author at: Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA, 90095-1592, USA.
E-mail address: pdc@seas.ucla.edu (P.D. Christofides).

https://doi.org/10.1016/j.compchemeng.2023.108498
Received 29 August 2023; Received in revised form 11 October 2023; Accepted 27 October 2023

and ultimately causing the high-speed centrifuges to burnout (Kushner, 2013). Another noteworthy incident involves the cyberattack on the Ukrainian power grid in 2015. During this event, hackers infiltrated SCADA systems to remotely shutdown substations, resulting in power failures. A more recent occurrence took place in 2021, concerning the Colonial Pipeline, a major operator of fuel pipelines in the United States. This company fell victim to a ransomware attack, orchestrated by hackers who gained entry through the use of the DarkSide ransomware. The attackers proceeded to encrypt the networked communication of the pipeline, demanding a ransom payment in return for the decryption keys. Consequently, Colonial Pipeline had to cease its operations, resulting in interruptions to fuel distribution and causing notable financial loss. These examples underscore the imperative for robust cybersecurity protocols in OT infrastructures.

Extensive research efforts continue to focus on various domains, such as the design of backup controllers in a two-tier safety-performance control architecture (Chen et al., 2020), the creation of machine learning-based cyberattack detectors (Huang et al., 2007; Omar et al., 2013; Agrawal and Agrawal, 2015; Wu et al., 2018), the recovery of process states following a cyberattack (Wu et al., 2020), the development of cyberattack-resilient controllers (Durand, 2018; Durand and Wegener, 2020), and encrypted control (Suryavanshi et al., 2023). However, this research aims to integrate some of these approaches, particularly machine-learning based cyberattack detection in a two-tier encrypted control architecture, to create a robust and cyber-secure control scheme applicable to nonlinear processes.

Networked communication lines are vulnerable to cyberattacks when data is transmitted in its regular plaintext form. To address this, encryption emerges as a solution, effectively safeguarding data during its transfer. Within control systems, data serves as the foundation for computing control inputs. While encryption offers enhanced security, it also introduces limitations, allowing only linear computations—a drawback that can hamper the utilization of advanced controllers like model predictive control (MPC) in complex systems characterized by nonlinear dynamics.

MPC ensures closed-loop stability (confinement of system states within a level set of the control Lyapunov function), optimizes critical performance metrics, handles multi-input multi-output scenarios, and manages constraints on system states and inputs. These advantages stem from the deployment of a mathematical model to predict future behavior and consequently optimize control inputs by minimizing a cost function. However, for the application of MPC, decryption becomes necessary to provide the required measurements for prediction and optimization at the end of the controller. While a linear control law provides the ability to calculate control inputs in an encrypted space, eliminating the need for decryption and ensuring a more secure approach, the advantages of nonlinear model predictive control cannot be ignored. Moreover, a delicate balance exists between improving system cybersecurity and enhancing closed-loop performance. Thoughtful assessments are necessary, taking into account the improvement achieved with the nonlinear controller, the level of cybersecurity in the process setting, and, most crucially, the adherence to the necessary physical safety standards for the process. Similarly, the selection of a nonlinear controller, even with the aim of improving closed-loop control performance, might not be justified if it increases the vulnerability of the system to cyberthreats.

To reconcile the benefits of both paradigms, we propose an encrypted two-tier control architecture coupled with ML-based cyberattack detection. In this setup, the lower tier is composed of an encrypted linear control scheme capable of calculating control inputs within an encrypted space, eliminating the requirement for decryption in the network. This self-contained lower tier is capable of independently stabilizing the system. Conversely, the upper tier comprises an encrypted nonlinear controller (e.g., MPC) that receives encrypted signals which are decrypted to plaintext upon arrival to compute control inputs. The computed plaintext control inputs are subsequently encrypted before transmission to the actuator. It is crucial to emphasize that the plaintext data received by the MPC and the computed plaintext control inputs are both susceptible to cyberattacks in the networked upper tier.

However, with ML-based cyberattack detection integrated in the encrypted control architecture, when a cyberattack is detected, the compromised upper tier is deactivated, and exclusively the secure and stabilizing lower tier is utilized to regain system stability. This approach enables us to amalgamate the strengths of cyber-secure encrypted linear control and advanced nonlinear control, to create a cyber-secure, advanced nonlinear control scheme that fortifies the system against cyberattacks. Beyond ML-based cyberattack detection, alternative detection strategies can be considered. These include a reachable set-based detection scheme as explored in the work of Narasimhan et al. (2023), where a set is created that includes all possible states that a system can reach or achieve under specific control inputs and initial conditions. Deviations from these expected states could indicate a potential cyberattack. However, this method is restricted to linear systems. Another approach involves employing a controller switching technique, wherein controller-observer parameter switching occurs between nominal system parameters and attack-sensitive system parameters to facilitate attack detection (Narasimhan et al., 2022a,b). However, this method may fail to detect intelligent cyberattacks which are designed to avoid detection by conventional metrics such as residual errors. However, this study only focuses on intelligent cyberattacks, which are discussed in Section 4.

In the previous work of Suryavanshi et al. (2023), it was assumed that the computing unit responsible for decrypting states and computing control inputs is cybersecure. However, in this current study, we have developed a more robust control framework. Even if the computing unit is not secure and comes under a cyberattack, our control system logic deactivates the upper-tier controller and solely relies on the encrypted lower-tier controller to stabilize the system. This lower-tier control is linear and operates within an encrypted space and does not share access to public and private keys with the computing unit, unlike the upper-tier controller, which is nonlinear. Consequently, even in scenarios where the environment for computing control inputs is not considered cybersecure, our proposed control framework can be used to enhance cybersecurity. As an alternative to a secure encrypted lower tier, a locally secure tier with backup sensors could potentially be employed (Chen et al., 2020). However, employing an encrypted lower tier ensures a continuous and seamless flow of encrypted network communication, which can solely be accessed by authorized personnel equipped with the required private keys necessary for decryption. Consequently, this approach eliminates the necessity for secure local communication that is isolated from the network, which poses challenges in terms of access. This distinctive aspect underscores the novelty and significance of this research.

The subsequent sections of this paper are structured as follows: in Section 2, we present the notation, describe the class of systems employed, explain the cryptosystem applied for encryption, and the implications of quantization; In Section 3 we elaborate on the architecture design of the encrypted two-tier control, outline the formulation of both the encrypted lower tier and upper tier, followed by a stability analysis to identify sources of errors in the control framework and set bounds to it; In Section 4 we describe the various launched cyberattacks and the machine-learning-based cyberattack detector; in Section 5, we showcase the application of the proposed control scheme on a nonlinear chemical process network, explain the important points to be considered while implementing the control framework in nonlinear systems, and put forth the computational load arising from the incorporation of ML-based detection within the encrypted control scheme.

2. Preliminaries

2.1. Notation

The symbol $\|\cdot\|$ represents the Euclidean norm of a vector. The transpose of the vector x is denoted by $x^\top$. $\mathbb{R}$, $\mathbb{Z}$, and $\mathbb{N}$ denote

the sets of real numbers, integers, and natural numbers, respectively. Moreover, the notations $\mathbb{Z}_M$ and $\mathbb{Z}_M^*$ are used to represent the additive and multiplicative groups of integers modulo M, correspondingly. The operation of subtracting sets is indicated by the symbol "\", such that $A\backslash B$ denotes the set of elements present in A but not in B. A function denoted as $f(\cdot)$ is categorized as belonging to class C^1 if it possesses continuous differentiability within its domain. A function $\alpha : [0, a) \to [0, \infty)$ is categorized within the class $\mathcal{K}$ when it is strictly increasing and $\alpha(0) = 0$. The term $\mathrm{lcm}(i, j)$ indicates the least common multiple of the integers i and j. The term $\gcd(i, j)$ indicates the greatest common divisor, which identifies the highest positive integer that divides i and j without any remainder.

2.2. Class of systems

The focus of this research is on nonlinear continuous-time systems featuring multiple inputs and multiple outputs (MIMO), characterized by a collection of nonlinear first-order ordinary differential equations (ODEs) of the form,

$$\dot{x} = F(x, u_{t1}, u_{t2}) = f(x) + g_1(x)u_{t1} + g_2(x)u_{t2} \tag{1}$$

The system is described by a state vector $x = [x_1, x_2, \ldots, x_n] \in \mathbb{R}^n$, a lower-tier control input vector $u_{t1} \in \mathbb{R}^{m_1}$ and an upper-tier control input vector $u_{t2} \in \mathbb{R}^{m_2}$. The system inputs, denoted as u_{t1} and u_{t2}, are bounded by their respective sets $U_1 \subset \mathbb{R}^{m_1}$ and, $U_2 \subset \mathbb{R}^{m_2}$, where $U_1 := \{u_{t1} \in U_1 | u_{t1_{\min,i}} \le u_{t1_i} \le u_{t1_{\max,i}}, \forall; i = 1, 2, \ldots, m_1\}$ and $U_2 := \{u_{t2} \in U_2 | u_{t2_{\min,i}} \le u_{t2_i} \le u_{t2_{\max,i}}, \forall; i = 1, 2, \ldots, m_2\}$. The quantities $u_{t1_{\min,i}}$ and $u_{t1_{\max,i}}$ correspond to the lowest and highest thresholds for each controlled input in the lower tier, respectively. Similarly, the values $u_{t2_{\min,i}}$ and $u_{t2_{\max,i}}$ pertain to the minimum and maximum values allowed for each controlled input in the upper tier. The functions $f(\cdot)$, $g_1(\cdot)$, and $g_2(\cdot)$ are assumed to be sufficiently smooth vector functions, respectively. For the purpose of simplicity without loss of generality, we introduce the assumption that $f(0) = 0$, effectively treating the origin as a steady state of Eq. (1). For the sake of convenience, we establish the initial time as zero ($t_0 = 0$). Furthermore, the domain of continuous functions that map the interval $[a, b]$ to $\mathbb{R}^n$ is designated as $C([a, b], \mathbb{R}^n)$. Additionally, we define the set $S(\Delta)$ as the assortment of piece-wise constant functions characterized by a period of Δ.

2.3. Paillier cryptosystem

In this research, we employ the Paillier cryptosystem (Paillier, 1999) to implement encryption and decryption procedures on state measurements of the process (denoted as x) as well as control inputs (represented as u_{t1} and u_{t2}). More importantly, we leverage the semi-homomorphic property of additive homomorphism within the Paillier cryptosystem to conduct linear additive operations within an encrypted space in the lower tier. Similar to numerous other encryption methods, the Paillier cryptosystem's functionality centers on the encryption of plaintext data in the format of natural numbers. The encryption procedure is initiated with the creation of public and private keys. Within the Paillier cryptosystem, integer messages are encrypted to ciphertexts by utilizing the public key during the encryption process. In contrast, the private key facilitates the decryption of ciphertexts, to recover the initial integer messages. The public and private keys are generated as per the following steps:

1. Choose two large prime integers (p and q) randomly, ensuring they meet the requirement $\gcd(pq, (p-1)(q-1)) = 1$.
2. Compute the outcome of multiplying these integers, indicated as $M = pq$.
3. Choose an arbitrary integer g in a manner that $g \in \mathbb{Z}_{M^2}$, with $\mathbb{Z}_{M^2}$ denoting the multiplicative group of integers modulo M^2.
4. Compute $\lambda = \mathrm{lcm}(q-1, p-1)$.
5. Specify $\bar{L}(x) = (x-1)/M$.
6. Verify whether the subsequent modular multiplicative inverse is present:
   $u = (\bar{L}(g^\lambda \bmod M^2))^{-1} \bmod M$.
7. Should the inverse not exist, revisit step 3 and opt for an alternate value of g. If the inverse exists, we acquire the public key (M, g) and the private key (λ, u).

Upon acquiring the keys, the public key is disseminated to the intended recipients responsible for carrying out the encryption procedure. Similarly, the private key is shared with the authorized recipients responsible for decrypting the data. Encryption is performed as follows:

$$E_M(m, r) = c = g^m r^M \bmod M^2 \tag{2}$$

where r is a randomly selected integer from the set $\mathbb{Z}_M$, and c represents the ciphertext achieved through the encryption of m. The decryption procedure for the ciphertext $c \in \mathbb{Z}_{M^2}$ is executed in the subsequent manner:

$$D_M(c) = m = \bar{L}(c^\lambda \bmod M^2)u \bmod M \tag{3}$$

2.4. Quantization

To utilize the Paillier cryptosystem, it becomes imperative to represent the data to be encrypted as natural numbers, a subset designated as $\mathbb{Z}_M$. However, the signal measurements before encryption are available in the form of floating-point numbers. Consequently, we use the process of quantization to map these floating-point numbers into elements of the set $\mathbb{Z}_M$. To construct this mapping, we use signed fixed-point numbers represented in binary form. The parameters of quantization, namely l_1 and d, signify the total count of bits (integer and fractional) and the number of fractional bits, respectively. Employing these quantization parameters, we create a set denoted as $\mathbb{Q}_{l_1,d}$. This set encompasses rational numbers spanning from -2^{l_1-d-1} to $2^{l_1-d-1} - 2^{-d}$, with each rational number separated by a step of 2^{-d}. A rational number q that resides within the $\mathbb{Q}_{l_1,d}$ set can be articulated as $q \in \mathbb{Q}_{l_1,d}$, where, $\exists \beta \in \{0, 1\}^{l_1}$ and $q = -2^{l_1-d-1}\beta_{l_1} + \sum_{i=1}^{l_1-1} 2^{i-d-1}\beta_i$. In order to map a real number data point a onto the $\mathbb{Q}_{l_1,d}$ set, we employ the function $g_{l_1,d}$, illustrated by the equation,

$$\begin{aligned} &g_{l_1,d} : \mathbb{R} \to \mathbb{Q}_{l_1,d} \\ &g_{l_1,d}(a) := \arg\min_{q \in \mathbb{Q}_{l_1,d}} |a - q| \end{aligned} \tag{4}$$

to acquire the nearest quantized rational number to a specific real number data point. After this, the quantized data is converted into a collection of integers via a one-to-one (bijective) mapping referred to as $f_{l_2,d}$, as described in the work of Darup et al. (2017). This mapping guarantees that the quantized data undergoes a transformation that places it within a subset of the message space $\mathbb{Z}_M$. The one-to-one mapping can be defined as follows:

$$\begin{aligned} &f_{l_2,d} : \mathbb{Q}_{l_1,d} \to \mathbb{Z}_{2^{l_2}} \\ &f_{l_2,d}(q) := 2^d q \bmod 2^{l_2} \end{aligned} \tag{5}$$

The encryption process involves encrypting integer plaintext messages using the set $Z_{2^{l_2}}$, and the resulting ciphertexts can be decrypted back into the same set $Z_{2^{l_2}}$. Once the upper-tier controller and actuator receive the encrypted signals, the ciphertexts undergo decryption to extract integer plaintext messages that represent quantized states and inputs, respectively. Consequently, it becomes essential to remap these decrypted plaintext messages back to the set $\mathbb{Q}_{l_1,d}$. The inverse mapping, denoted as $f_{l_2,d}^{-1}$, is defined as follows:

$$f_{l_2,d}^{-1} : \mathbb{Z}_{2^{l_2}} \to \mathbb{Q}_{l_1,d} \tag{6}$$

$$f_{l_2,d}^{-1}(m) := \frac{1}{2^d} \begin{cases} m - 2^{l_2} & \text{if } m \ge 2^{l_2-1} \\ m & \text{otherwise} \end{cases} \tag{7}$$

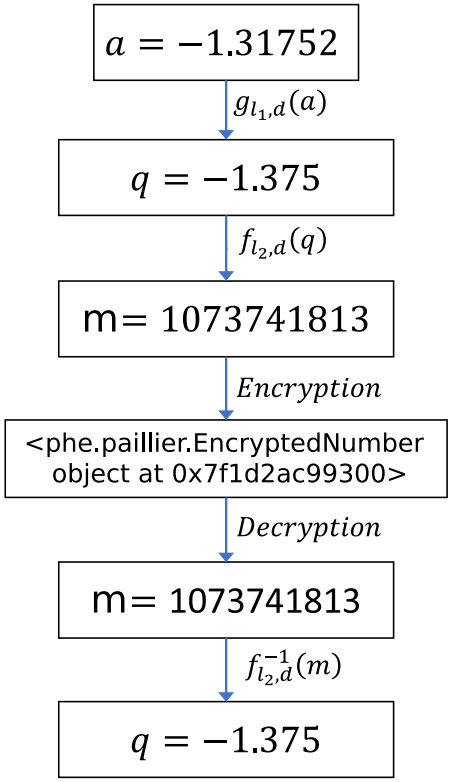

Fig. 1. Visualization of the encryption–decryption process and effect of quantization when applied to a floating-point real number.

To illustrate the process of encryption and decryption, we can refer to the example shown in Fig. 1. For this specific instance, the selected quantization parameters are as follows: $d = 3$, $l_1 = 18$, and $l_2 = 30$. Let us consider the rational number $a = -1.31752$. The impact of quantization is demonstrated in Fig. 1, where the quantization error, $|a - q| = 0.05748$, is evident.

3. Design of the encrypted two-tier control architecture

In the closed-loop framework of the encrypted two-tier control architecture, illustrated in Fig. 2, the signals, $x_1(t)$ and $x_2(t)$ are transmitted from the sensors to the lower and upper tier, respectively, for the purpose of computing control inputs. The lower and upper tier correspond to the encrypted network tiers 1 and 2 respectively, in Fig. 2. These signals $x_1(t)$ and $x_2(t)$ undergo encryption using public keys 1 and 2 respectively before they are transmitted to the lower tier consisting of a set of encrypted proportional–integral (PI) controllers and the upper tier comprising a model predictive controller (MPC), respectively. These two tiers operate independently for control input computations, utilizing distinct public and private keys for signal encryption and decryption. Further, both tiers manipulate a distinct set of control inputs, eliminating any concerns related to balancing control signals among actuators. Once the lower tier receives the encrypted data, denoted as c_1, it performs control input calculations within an encrypted space, without decryption, employing additive homomorphic operations. The encrypted control input c_1' is then transmitted to the actuator, where it undergoes decryption using private key 1 to yield the quantized control input $\hat{u}_1(t)$. Concurrently, the upper tier decrypts the ciphertext c_2 and employs the quantized states $\hat{x}_2(t)$ to determine the control input. These quantized states are used to initialize the process model within the MPC at the time t. Following this, the MPC calculates the optimized control inputs $u_2(t)$, which undergo encryption before being transmitted to the actuator. Upon receipt of the encrypted control input c_2', the actuator decrypts it using private key 2, leading to the quantized input $\hat{u}_2(t)$, which is then applied to the process. The presented architecture introduces two potential points within the upper tier where cyberattacks could be initiated: one by manipulating the decrypted state values received by the MPC, and the other by manipulating the control inputs computed by the MPC before encryption. To counteract this vulnerability, an ML-based detector is incorporated at the process site. It intercepts sensor signals prior to their encryption and transmission to the network, thereby ensuring its security. Its role is to detect cyberattacks and subsequently reconfigure the control system in the event of cyberattack detection. This reconfiguration involves deactivating the compromised upper tier and relying solely on the secure, encrypted lower tier to restore the desired closed-loop behavior.

Remark 1. The encrypted data, in the form of ciphertexts, could potentially be subject to manipulation by an attacker. However, due to the encryption, the attacker gains no information about the process states or the system stability. Any attempts to manipulate the encrypted data would lead to significant deviations from actual values. The manipulated encrypted data after decryption could yield infeasible values for certain states, and some control inputs could fall outside the actuation bounds. Such alterations have the potential to destabilize the system, and they can be easily identified by imposing constraints on the control Lyapunov function, eliminating the need for advanced detection techniques. However, in this research, we focus on intelligent cyberattacks that do not force the system out of its stability region. These attacks require the attacker to possess some knowledge about the system and its states, information that can only be obtained through decryption of the states and computation of control inputs before encryption. Therefore, our discussion is centered around these scenarios. Further details regarding the types of cyberattacks launched are provided in Section 4. Additionally, as a proactive measure, a backup control system can be integrated into this design, operating in isolation from any network, to address potential cyberattacks aimed at manipulating encrypted data.

The presented design of the closed-loop system introduces two types of errors. Initially, there is a quantization error due to the mapping of state data from $\mathbb{R}$ to $\mathbb{Q}_{l_1,d}$ within the sensor–controller communication link. Furthermore, the controller–actuator communication link contributes a control input quantization error as the control input is mapped from a set of real numbers $\mathbb{R}$ to $\mathbb{Q}_{l_1,d}$. Both of these quantization errors are constrained and can be characterized via the mapping equation specified in Eq. (4), thereby ensuring that:

$$|x_j(t) - \hat{x_j}(t)| \leq 2^{-d-1} \tag{8a}$$

$$|u_k(t) - \hat{u_k}(t)| \leq 2^{-d-1} \tag{8b}$$

where d is the quantization parameter used for mapping in Eq. (4), while j and k represent the jth state and kth control input, respectively. Taking into account the impact of quantization-induced input errors, the dynamical model under two-tier control architecture employing the nonlinear system of Eq. (1) can be expressed as follows:

$$\begin{aligned}\dot{x} = F(x, \hat{u}_{t1}, \hat{u}_{t2}) &= f(x) + g_1(x)\hat{u}_{t1} + g_2(x)\hat{u}_{t2} \\ &= f(x) + g_1(x)(u_{t1} + e_{t1}) + g_2(x)(u_{t2} + e_{t2})\end{aligned} \tag{9}$$

where $e_{t1} = \hat{u}_{t1}(t) - u_{t1}(t)$, $e_{t2} = \hat{u}_{t2}(t) - u_{t2}(t)$ and

$$|e_{ti}| \leq 2^{-d-1} \text{ where } i = \{1, 2\} \tag{10}$$

Also, an additional error will be present in the applied control input, as the controller receives $\hat{x}$ instead of the true state x. This error will be confined by the underlying equation, using the local Lipschitz property, where $L_1 > 0$:

$$|\Phi(\hat{x}) - \Phi(x)| \leq L_1|\hat{x} - x| \leq L_1 2^{-d-1} \tag{11}$$

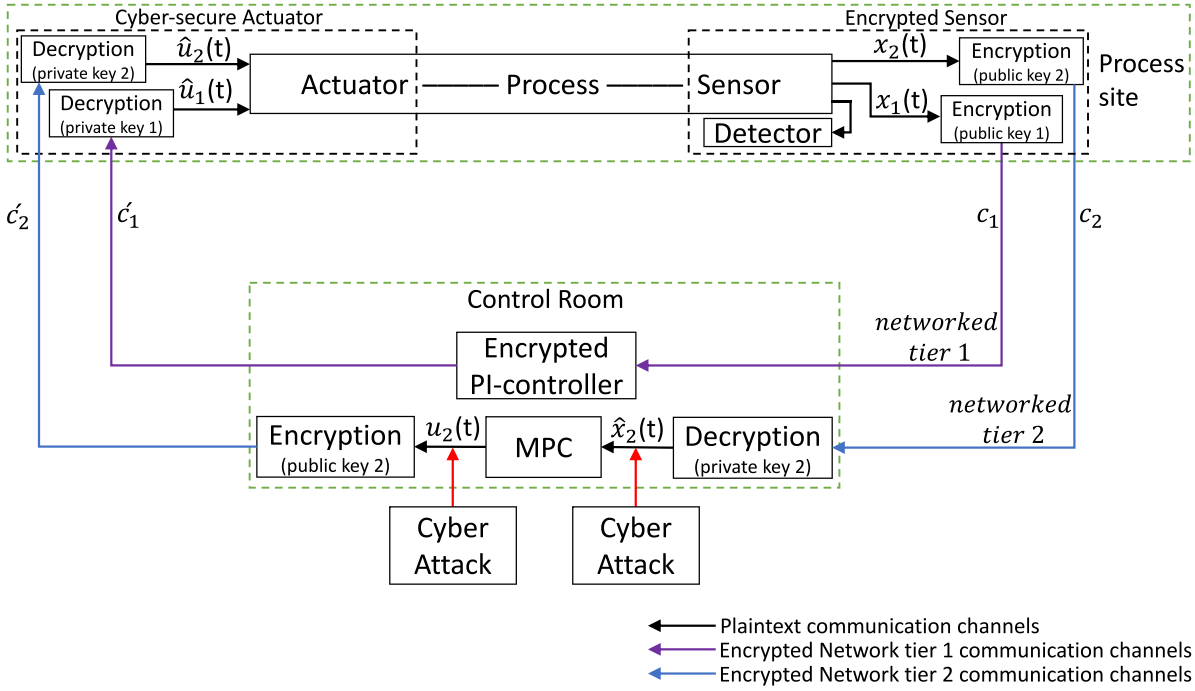

Fig. 2. Illustration of a two-tier encrypted control scheme.

Remark 2. Quantization error arises when the desired value to be quantized is not exactly found within the $\mathbb{Q}_{l_1,d}$ set, which comprises quantized values defined by the quantization parameter d. The interval between elements in this set is 2^{-d}. To ascertain the upper limit of this error, let us consider the quantization of a value, x_1. We assume that x_1 falls within the range of y_1 and $y_1 + 2^{-d}$, where y_1 and $y_1 + 2^{-d}$ signify quantized values within $\mathbb{Q}_{l_1,d}$. The quantization procedure involves evaluating the absolute difference between x_1 and y_1 in comparison to that between x_1 and $y_1 + 2^{-d}$. When the distance between x_1 and y_1 is less than the distance between x_1 and $y_1 + 2^{-d}$, x_1 is matched with y_1. Alternatively, it is matched with $y_1 + 2^{-d}$. Subsequently, the quantization error is confined within half of the resolution, $\frac{|y_1+2^{-d}-y_1|}{2} = 2^{-d-1}$. This implies that the maximum difference between the quantized value $\hat{x}_1$ and the actual value x_1 is 2^{-d-1}. Thus, selecting a larger quantization parameter, $d \to \infty$, results in a negligible error due to quantization.

3.1. Lower-tier encrypted control system

Within the encrypted two-tier control framework, we assume the existence of a feedback controller in the lower tier, represented as $u_{t1} = \Phi(x) \in U_1$, that can attain exponential stability at the origin of the nominal closed-loop system of Eq. (1), with $u_{t2} \equiv 0$. This signifies the presence of a C^1 control Lyapunov function $V(x)$ for which the subsequent inequalities are valid across all $x \in \mathbb{R}^n$ within an open region D surrounding the origin:

$$c_1|x|^2 \leq V(x) \leq c_2|x|^2, \tag{12a}$$

$$\frac{\partial V(x)}{\partial x} F(x, \Phi(x), 0) \leq -c_3|x|^2, \tag{12b}$$

$$\left|\frac{\partial V(x)}{\partial x}\right| \leq c_4|x| \tag{12c}$$

where c_1, c_2, c_3 and c_4 are positive constants. For the nonlinear system described by Eq. (1), the region of closed-loop stability can be defined as a level set of the control Lyapunov function V. This stability domain, labeled as Ω_ρ, is defined by $\Omega_\rho := \{x \in D | V(x) \leq \rho\}$, where $\rho > 0$. Hence, originating from any initial condition within Ω_ρ, the applied control input, $\Phi(x)$ guarantees that the system state trajectory, under closed-loop conditions, remains confined within Ω_ρ.

To perform computations in an encrypted space, classical controllers using linear mathematical operations are used to compute control inputs. Specifically, a set of proportional–integral controllers are used. The formula is given as:

$$u(t_k) = K_{c_i}\left(e_i(t_k) + \frac{1}{\tau_i}\int_0^{t_k} e_i(\tau)d\tau\right), \; e_i(t_k) = y_{sp}(t_k) - y_i(t_k) \tag{13}$$

Using the recursive rule to approximate the integral term, the overall controller equation is reformulated using only linear mathematical operations:

$$\begin{aligned} u_{t1_i}(t_k) &= K_{c_i} e_i(t_k) + I_{t_k} \\ &= K_{c_i} e_i(t_k) + K'_{c_i} e_i(t_k) + I_{t_{k-1}} \end{aligned} \tag{14}$$

where t_k and t_{k-1} represent the sampling instances k and $k-1$, respectively. u_{t1_i} represents the ith control input in the lower tier, $y_{sp}(t_k)$ and $y_i(t_k)$ represent the set point and state measurement at time t_k, respectively. K_{c_i} and K'_{c_i} represent the gains of the proportional and integral terms, respectively. I_{t_k} represents the integral control action at time t_k. At $k = 0$, I_{t_0} is assumed to be 0.

3.2. Upper-tier encrypted model predictive control system

This section formulates the feedback LMPC used in the upper tier of the closed-loop design for the nonlinear system described by Eq. (1). Although the LMPC does not compute the control inputs for the lower tier, it estimates their values using the lower tier control law, $u_{t1}(t) = \Phi(\tilde{x}(t))$. This estimation results in a more accurate prediction of the future states of the system, by accounting in the lower-tier control inputs. These predicted state values are used to calculate the LMPC cost function. Accordingly, the upper-tier control inputs that minimize the cost function are computed. Control actions are applied to the nonlinear system using a sample-and-hold approach with a sampling period of Δ (Heidarinejad et al., 2012; Mhaskar et al., 2006). The proposed MPC is formulated in the subsequent manner:

$$\mathcal{J} = \min_{u_{t2} \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), \Phi(\tilde{x}(t)), u_{t2}(t)) \, \mathrm{d}t \tag{15a}$$

$$\text{s.t.} \quad \dot{\tilde{x}}(t) = F(\tilde{x}(t), \Phi(\tilde{x}(t)), u_{t2}(t)) \tag{15b}$$

$$u_{t2}(t) \in U_2, \ \forall\ t \in [t_k, t_{k+N}) \tag{15c}$$

$$\tilde{x}(t_k) = \hat{x}(t_k) \tag{15d}$$

$$\dot{V}(\hat{x}(t_k), \Phi(\hat{x}(t_k)), u_{t2}(t_k)) \leq \dot{V}(\hat{x}(t_k), \Phi(\hat{x}(t_k)), 0), \ \text{if } \hat{x}(t_k) \in \Omega_{\hat{\rho}} \backslash \Omega_{\rho_{\min}} \tag{15e}$$

$$V(\tilde{x}(t)) \leq \rho_{\min}, \ \forall\ t \in [t_k, t_{k+N}), \ \text{if } \hat{x}(t_k) \in \Omega_{\rho_{\min}} \tag{15f}$$

The predicted state trajectory of the LMPC process model is represented as $\tilde{x}$. The quantized states, $\hat{x}$, serve as the initial conditions for the LMPC process model to predict the state trajectories. The number of sampling periods within the prediction horizon is represented as N. The LMPC algorithm computes the optimal input sequence $u^*_{t2}(t|t_k)$ for the entire prediction horizon $t \in [t_k, t_{k+N})$ but transmits only the first input of this sequence to the actuator for application to the system within the interval $t \in [t_k, t_{k+1})$. The rationale behind predicting state trajectories for extended durations compared to the control input application period by the actuator is to optimize the existing control inputs. This optimization aims to minimize the control cost function not only within the current sampling period but also over the prediction horizon, thereby enhancing overall performance.

The encrypted LMPC method employs a sequence of specific actions: it uses quantized states $\hat{x}$ to predict the trajectory of the system states as per Eq. (15b), which is used to integrate the cost function of Eq. (15a) to calculate optimized control inputs for the entire prediction horizon. The actuator applies only the control inputs of the first sampling period, and this process is iterated at each sampling period. Eq. (15c) represents the constraints imposed on the control inputs. The constraint in Eq. (15d) uses the quantized states (after decryption) to initialize the plant model described in Eq. (15b). If the state $x(t_k)$ at time t_k lies within the set $\Omega_{\hat{\rho}} \setminus \Omega_{\rho_{\min}}$, where $\rho_{\min}$ represents a level set of V in proximity to the origin, the Lyapunov constraint outlined in Eq. (15e) ensures that the time-derivative of the control Lyapunov function of the closed-loop system under the two-tier control scheme is less than or equal to the time-derivative of the control Lyapunov function when the system is controlled by only the lower tier. When the closed-loop state $x(t_k)$ enters $\Omega_{\rho_{\min}}$, the constraint detailed in Eq. (15f) ensures that this state remains within $\Omega_{\rho_{\min}}$.

3.3. Lower-tier stability under encryption

Given the occurrence of quantization errors in the links between sensors and controllers, as well as controllers and actuators, it becomes imperative to delineate a region of closed-loop stability, denoted as $\Omega_{\hat{\rho}}$, which is encompassed within the broader Ω_{ρ} (specifically, $\hat{\rho} < \rho$). The subsequent theorem establishes that the encrypted lower-tier controller $\Phi(\hat{x}) \in U_1$ can achieve exponential stability at the origin for the nonlinear system introduced in Eq. (9).

Theorem 1. *Let us consider the nonlinear system introduced in Eq. (9), which can be represented as $\dot{x} = F(x, \hat{u}_{t1}, 0)$ when exclusively under lower-tier encrypted control. The initial state is $x_0 \in \Omega_{\hat{\rho}}$, and the stabilizing control law is denoted as $u_{t1} = \Phi(x) \in U_1$. Consequently, the equilibrium point of the closed-loop system derived from Eq. (9) through encrypted control becomes practically stable for all $x_0 \in \Omega_{\hat{\rho}}$. In this context, the closed-loop state $x(t)$ remains within Ω_{ρ} for all instances, and the ensuing inequalities remain valid:*

$$\dot{V} \leq -c_5|x|^2 \quad \forall |x| \geq \frac{c_4(L_1+1)2^{-d-1}}{c_3\theta} = \mu \tag{16a}$$

$$\limsup_{t\to\infty} |x| \leq b \tag{16b}$$

where d is the quantization parameter, $c_3, c_4, L_1 > 0$, b is a positive constant (which can be expressed as a class $\mathcal{K}$ function of μ), $0 < \theta < 1$ and $c_5 = (1-\theta)c_3$.

Proof. Based on the nonlinear system of Eq. (9), the time-derivative of V can be written as:

$$\begin{aligned}
\dot{V} &= \frac{\partial V}{\partial x} F(x, \hat{u}_{t1}, 0) \\
&= \frac{\partial V}{\partial x} F(x, u_{t1} + e_1, 0) \\
&= \frac{\partial V}{\partial x} F(x, \Phi(\hat{x}) + e_1, 0) \\
&= \frac{\partial V}{\partial x} F(x, \Phi(\hat{x}) + e_1, 0) - \frac{\partial V}{\partial x} F(x, \Phi(x), 0) + \frac{\partial V}{\partial x} F(x, \Phi(x), 0)
\end{aligned} \tag{17}$$

Based on Eq. (12b), it follows that

$$\begin{aligned}
\dot{V} &\leq \frac{\partial V}{\partial x} F(x, \Phi(\hat{x}) + e_1, 0) - \frac{\partial V}{\partial x} F(x, \Phi(x), 0) - c_3|x|^2 \\
&= \frac{\partial V}{\partial x} \big(f(x) + g_1(x)(\Phi(\hat{x}) + e_1)\big) - \frac{\partial V}{\partial x} \big(f(x) + g_1(x)(\Phi(x))\big) - c_3|x|^2 \\
&= \frac{\partial V}{\partial x} \big(f(x) + g_1(x)(\Phi(\hat{x}) + e_1) - f(x) - g_1(x)(\Phi(x))\big) - c_3|x|^2 \\
&= \frac{\partial V}{\partial x} \big(g_1(x)(\Phi(\hat{x}) - \Phi(x))\big) + \frac{\partial V}{\partial x} g_1(x) e_1 - c_3|x|^2
\end{aligned} \tag{18}$$

Applying the inequalities of Eqs. (12c), (10) and (11), it follows that

$$\begin{aligned}
\dot{V} &\leq c_4|x|L_1 2^{-d-1} + c_4|x|2^{-d-1} - c_3|x|^2 \\
&= -c_3|x|^2 + c_4|x|(L_1+1)2^{-d-1} \\
&= -(1-\theta)c_3|x|^2 - \theta c_3|x|^2 + c_4|x|(L_1+1)2^{-d-1}
\end{aligned} \tag{19}$$

Therefore, if the condition of Eq. (16a) on $|x|$ is satisfied i.e., $|x| \geq \frac{c_4(L_1+1)2^{-d-1}}{c_3\theta} = \mu$, it follows that

$$\begin{aligned}
\dot{V} &\leq -(1-\theta)c_3|x|^2 \leq 0 \\
&\leq -c_5|x|^2 \leq 0
\end{aligned} \tag{20}$$

where $c_5 = (1-\theta)c_3$. Thus, based on Eq. (20), we have that $\dot{V}$ is negative for all $x \in \Omega_{\hat{\rho}}$ that satisfy the condition of Eq. (16a).

Given that $\Omega_{\hat{\rho}}$ is a level set of V and its derivative, $\dot{V}$, is negative for all $x \in \Omega_{\hat{\rho}}$, it can be inferred that the state of the closed-loop system, denoted as $x(t)$, remains within $\Omega_{\hat{\rho}}$ throughout all time. Moreover, referencing Theorem 4.18 in Khalil (2002), it can be deduced that:

$$\limsup_{t\to\infty} |x(t)| \leq b \tag{21}$$

Hence, as the quantization parameter $d \to \infty$, following the definition of μ from Eq. (16a), $\mu \to 0$ and, therefore, the ultimate bound approaches zero, proving that larger values of the quantization parameter d results in a smaller error between the state and input trajectories of the encrypted control system and the non-encrypted control system. This proves that the closed-loop states of the nonlinear system of Eq. (9) are ultimately bounded under the stabilizing controller $u_{t1} = \Phi(\hat{x}) \in U_1$ for sufficiently large d. □

3.4. Two-tier stability under encryption

Theorem 2. *Taking into consideration the two-tier encrypted control architecture for the system of Eq. (9), we examine its behavior within the context of the closed-loop encrypted LMPC design detailed in Eq. (15) for the upper tier. This design relies on a stabilizing lower-tier controller denoted as $u_{t1} = \Phi(\hat{x}) \in U_1$, which adheres to the inequalities outlined in Eq. (12). Furthermore, we assume that the initial state x_0 resides within the region $\Omega_{\hat{\rho}}$. For the purpose of our analysis, we introduce $\Delta > 0$, $\epsilon_w > 0$, and parameters $\hat{\rho} > \rho_{\min} > \rho_s$ that fulfill the following conditions.*

$$\begin{aligned}
&-\frac{c_3}{c_2}\rho_s + L'_x M_F \Delta + L'_u \delta \leq -\epsilon_w \\
&\rho_{\min} = \max\{V(x(t+\Delta)) | V(x(t)) \leq \rho_s\}
\end{aligned} \tag{22}$$

Then, the closed-loop state $x(t)$ remains bounded in $\Omega_{\hat{\rho}}$ and is ultimately bounded in $\Omega_{\rho_{\min}}$.

Proof. Consider the state $x(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_s}$. The time-derivative of V under the control inputs calculated by the LMPC of Eq. (15) for the nonlinear system of Eq. (9) at t_k can be written as:

$$\begin{aligned}\dot{V} &= \frac{\partial V(x(t))}{\partial x} F\left(x(t), \Phi(x(t_k)) + e_{t1}, u_{t2}(t_k) + e_{t2}\right) \\ \dot{V} &= \frac{\partial V(x(t_k))}{\partial x} F\left(x(t_k), \Phi(x(t_k)), u_{t2}(t_k)\right) \\ &\quad + \frac{\partial V(x(t))}{\partial x} F\left(x(t), \Phi(x(t_k)) + e_{t1}, u_{t2}(t_k) + e_{t2}\right) \\ &\quad - \frac{\partial V(x(t_k))}{\partial x} F\left(x(t_k), \Phi(x(t_k)), u_{t2}(t_k)\right)\end{aligned} \tag{23}$$

for all $t \in [t_k, t_{k+1}]$. Here, e_{t1} and e_{t2} represent the error in the control inputs of the lower and upper tiers, respectively, due to quantization. Based on Eqs. (18) and (19), the error e_{t1} can be bounded by $(L_1 + 1)2^{-d-1}$. Similarly, the error e_{t2} can be bounded by $\eta 2^{-d-1}$. Based on the inequality of Eq. (12b), it follows from Eq. (23) that:

$$\begin{aligned}\dot{V} \leq -c_3|x(t_k)|^2 &+ \frac{\partial V(x(t))}{\partial x} F\left(x(t), \Phi(x(t_k)) + e_{t1}, u_{t2}(t_k) + e_{t2}\right) \\ &- \frac{\partial V(x(t_k))}{\partial x} F\left(x(t_k), \Phi(x(t_k)), u_{t2}(t_k)\right)\end{aligned} \tag{24}$$

In the encrypted LMPC, the constraint of Eq. (15e) ensures that, if $x(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_{\min}}$, then the closed-loop state is driven towards the origin at t_{k+1} (to a lower level set of V). Based on the fact that the errors $|e_{t1}|$ and $|e_{t2}|$ are bounded, using the Lipschitz condition and the inequality of Eq. (12a), it follows from Eq. (24) that:

$$\dot{V} \leq -\frac{c_3}{c_2}\rho_s + L'_x|x(t) - x(t_k)| + L'_{u1}(L_1 + 1)2^{-d-1} + L'_{u2}\eta 2^{-d-1} \tag{25}$$

where $L'_x, L'_{u1}, L'_{u2} > 0$. Due to the continuity of $x(t)\ \forall\, t \in [t_k, t_{k+1})$, we can write that $|x(t) - x(t_k)| \leq M_F \Delta\ \forall\, t \in [t_k, t_{k+1})$. Using this bound, it follows from Eq. (25) that:

$$\dot{V} \leq -\frac{c_3}{c_2}\rho_s + L'_x M_F \Delta + L'_u \delta \tag{26}$$

where $L'_u = \left(L'_{u1}(L_1+1) + L'_{u2}\eta\right)$ is a positive constant. $\delta = 2^{-d-1}$ is also a positive constant, dependent on the quantization parameter d selected. As evident, the magnitude of the error due to quantization, represented by the last term of Eq. (26), will be smaller as $d \to \infty$. Hence, selecting a higher quantization parameter is advisable whenever possible. Thus, if $-\frac{c_3}{c_2}\rho_s + L'_x M_F \Delta + L'_u \delta \leq -\epsilon_w$, then $\dot{V} \leq -\epsilon_w$ for any $x(t_k) \in \Omega_{\hat{\rho}} \setminus \Omega_{\rho_s}$. This establishes that, if the conditions of Eq. (22) are met, the state of the closed-loop system is always bounded in $\Omega_{\hat{\rho}}$, and it ultimately converges to $\Omega_{\rho_s} \subseteq \Omega_{\rho_{\min}}$ and remains there. □

4. Cyberattack types and machine learning-based detection

The upper-tier control system, where encrypted sensor signals are decrypted upon receipt and further transmitted to the MPC in decrypted form, could be susceptible to cyberattacks. Similarly, the control inputs computed by the MPC prior to encryption might also face vulnerability to cyber threats. These signals, transmitted in plaintext, could potentially be manipulated by an attacker if the control room responsible for decryption and encryption lacks full cyber–physical security.

In contrast, the lower tier receives encrypted signals and calculates control inputs within an encrypted space, transmitting them without decryption within the lower networked-tier. This approach ensures complete security, even if the control room receiving and transmitting the encrypted signals is not entirely secure, as the data remains encrypted throughout the networked communication within the lower tier. Also, as depicted in Fig. 2, where the encrypted network tiers 1 and 2 correspond to the lower and upper tiers, respectively, the lower tier does not necessitate sharing access to its public and private keys with the control room, in contrast to the upper tier. This distinction contributes to the enhanced cybersecurity of the lower tier, even in situations where the security of the control room might be compromised. Upon detecting a cyberattack, the upper-tier control system is disabled, while only the lower-tier control system remains operational. The latter is capable of stabilizing the system at its steady state.

4.1. Types of cyberattacks

Given the adaptability of intelligent cyberattacks to process and control system behaviors, it is assumed that these attacks possess the potency to access information regarding the stability region of the two-tier controlled process. The scope of cyberattacks in the encrypted two-tier control architecture typically encompasses manipulation of signal data, where data received by the MPC and the control inputs computed by it could potentially be subjected to tampering. This study addresses attacks directed at both the sensor signals received by the MPC and the control inputs computed by it.

In regular operational scenarios, decrypted sensor signals accurately reflect the true state data. However, if this data is tampered with, it can lead to control actions driving the process away from its steady state. Likewise, manipulation of control inputs can deviate process states by withholding the necessary control action. Intelligent cyberattacks are designed in a manner such that, when launched on sensor signals, the controller is capable of calculating an appropriate control action, within the actuation bounds, using the attacked state. Similarly, when launched on control inputs, the manipulated data avoids falling beyond actuation limits, thereby evading detection by conventional mechanisms. To address these challenges, advanced machine learning algorithms utilizing neural networks are used for cyberattack detection. Some commonly launched attacks are considered below.

4.1.1. Min–max cyberattack

Min–Max cyberattacks are specifically crafted to maximize destabilizing impact within the shortest timeframe while evading detection. To maintain their concealment from conventional detection methods, min–max attacks target the lower value of the following two conditions:

1. A window around equilibrium: This condition centers around a window encompassing the equilibrium point of the affected state(s), representing a range of realistic physical operational conditions.
2. Extreme state values: The second condition revolves around state values situated farthest from the equilibrium point, whether they are minimum or maximum values. The intention is to ensure that the system remains within the closed-loop stability region Ω_ρ.

By introducing attacks based on the aforementioned conditions, it is guaranteed that the state measurements received by the controller after manipulation remain inside the stability region delineated by the configured operational window. Furthermore, these attacks circumvent setting off any conventional detection alarms rooted in boundary values.

The formulation of the min–max attack is expressed in the following manner:

$$\bar{x}(t_i) = \min\left\{\arg\max_{x\in\mathbb{R}^n}\{V(x(t_i)) \leq \rho\}, \arg\max_{x\in\mathbb{R}^n}\{x(t_i) \in \chi\}\right\}, \quad \forall i \in [i_o, i_o + L_a] \tag{27a}$$

$$\bar{u}_{t2}(t_i) = \min\left\{\arg\max_{u_{t2}\in\mathbb{R}^{m_2}}\{V(x(t_i)) \leq \rho\}, \arg\max_{u_{t2}\in\mathbb{R}^{m_2}}\{u_{t2}(t_i) \in U_2\}\right\}, \quad \forall i \in [i_o, i_o + L_a] \tag{27b}$$

where ρ defines the region of the Lyapunov function $V(x)$ that characterizes the stability boundaries of the closed-loop system under the two-tier control architecture. The notation $\chi = \{x_l \leq x \leq x_u\}$ represents the desired operating range for the system states, where x represents the compromised sensor signals to be received by the MPC after decryption at each time step. The value i_o marks the time when the attack is introduced, and L_a denotes the duration of the attack in terms of sampling periods. Similarly, the symbol $u_{t2}(t_i)$ signifies the control input that has been tampered with before encryption. The symbols $\bar{x}(t_i)$ and $\bar{u}_{t2}(t_i)$ correspond to the altered or manipulated values of the sensor signal and control input of the upper tier, respectively.

4.1.2. Replay cyberattack

In a replay attack, the attacker initially captures portions of the system output aligned with a regular operational state marked by substantial oscillations. Subsequently, the attacker intervenes to intercept and restore the present process state measurements to the previously recorded values. Replay attacks can be represented using the subsequent equations:

$$\bar{x}(t_i) = x(t_k),\ \forall k \in [k_o, k_o + L_a],\ \forall i \in [i_o, i_o + L_a] \tag{28a}$$

$$\bar{u}_{t2}(t_i) = u_{t2}(t_k),\ \forall k \in [k_o, k_o + L_a],\ \forall i \in [i_o, i_o + L_a] \tag{28b}$$

where $x(t_k)$ and $u_{t2}(t_k)$ are the true plant measurement and control input, respectively. L_a denotes the extent of the attack as measured in terms of sampling intervals. $\bar{x}(t_i)$ and $\bar{u}_{t2}(t_i)$ denote the sequence of replay attacks initiated at time t_{i_o} by duplicating prior plant measurements and control inputs recorded commencing from time t_{k_o}. As the previous plant outputs are derived from authentic closed-loop measurements and obtained via secure sensors, these state values are hypothesized to fall within the stability region and operating bounds. Consequently, by reproducing these values and reintroducing them into the controller, conventional detectors are unlikely to detect the anomaly.

4.1.3. False-data-injection cyberattack

False-data-injection (FDI) cyberattacks involve the insertion of fabricated information into authentic data. This intrusion does not necessitate familiarity with previous event data or system specifics. Introducing deceptive data such that $V(x) \leq \rho$ might not lead to system destabilization, but could merely modify its operational state based on the process dynamics, rendering them challenging to identify through conventional alarm threshold approaches. FDI attacks are represented as follows:

$$\bar{x}(t_i) = x(t_i) + \nu,\ \forall i \in [i_o, i_o + L_a] \tag{29a}$$

$$\bar{u}_{t2}(t_i) = u_{t2}(t_i) + \nu,\ \forall i \in [i_o, i_o + L_a] \tag{29b}$$

where $x(t_i)$ and $u_{t2}(t_i)$ are the true plant measurement and control input, respectively. ν represents the false data injected. L_a represents the length of the attack in terms of sampling periods. $\bar{x}(t_i)$ and $\bar{u}_{t2}(t_i)$ are the FDI attacks introduced from time t_{i_o} up to time $t_{i_o+L_a}$.

4.1.4. Sinusoidal cyberattack

Sinusoidal attack constitutes a form of cyberattack involving the introduction of a sinusoidal signal into authentic data. Due to the inherent periodic oscillations in a sinusoidal function, these attacks can be challenging to identify, as they lack the potential to destabilize the system while inducing substantial fluctuations. Moreover, their periodic pattern can evade standard detection mechanisms. Their representation can be expressed as follows:

$$\bar{x}(t_i) = x(t_i) + a\sin(2\pi k t_i),\ \forall i \in [i_o, i_o + L_a] \tag{30a}$$

$$\bar{u}_{t2}(t_i) = u_{t2}(t_i) + a\sin(2\pi k t_i),\ \forall i \in [i_o, i_o + L_a] \tag{30b}$$

where $x(t_i)$ and $u_{t2}(t_i)$ are the true plant measurement and control input, respectively. k and a are constants. L_a represents the length of the attack in terms of sampling periods. $\bar{x}(t_i)$ and $\bar{u}_{t2}(t_i)$ are the sinusoidal attacks introduced from time t_{i_o} up to time $t_{i_o+L_a}$.

4.1.5. Surge cyberattack

Surge cyberattack is a stealthy cyberattack that cannot be detected by conventional methods such as cumulative sum (CUMSUM). Surge attacks share similarities with min–max attacks in their initial behavior of maximizing disruptive impact over a brief interval before diminishing to a lower level. In our scenario, the initial duration of the surge, measured in sampling periods, is denoted as L_s and is chosen to be between 2 and 5 inclusive. This choice helps distinguish surge attacks from min–max attacks, as the surge exhibits distinct characteristics during its latter phase. After the sampling duration, L_s, a bounded noise is introduced to the genuine data, resembling the approach used in a false-data-injection attack. Their representation can be expressed as follows:

$$\bar{x}(t_i) = \min\left\{\arg\max_{x\in\mathbb{R}^n}\{V(x(t_i)) \leq \rho\}, \arg\max_{x\in\mathbb{R}^n}\{x(t_i) \in \chi\}\right\},\ \forall i \in [i_o, i_o + L_s] \tag{31a}$$

$$\bar{x}(t_i) = x(t_i) + \eta(t_i),\ \forall i \in (L_s, i_o + L_a] \tag{31b}$$

$$\bar{u}_{t2}(t_i) = \min\left\{\arg\max_{u_{t2}\in\mathbb{R}^{m_2}}\{V(x(t_i)) \leq \rho\}, \arg\max_{u_{t2}\in\mathbb{R}^{m_2}}\{u_{t2}(t_i) \in U_2\}\right\},\ \forall i \in [i_o, i_o + L_s] \tag{31c}$$

$$\bar{u}_{t2}(t_i) = u_{t2}(t_i) + \eta(t_i),\ \forall i \in (L_s, i_o + L_a] \tag{31d}$$

where $x(t_i)$ and $u_{t2}(t_i)$ are the true plant measurement and control input of the upper tier, respectively. The initial surge corresponds to Eqs. (31a) and (31c), while the subsequent noise addition is represented by Eqs. (31b) and (31d). $\eta_l \leq \eta(t_i) \leq \eta_u$ is the bounded noise added to the data following the initial surge. L_a represents the length of the attack in terms of sampling periods. $\bar{x}(t_i)$ and $\bar{u}_{t2}(t_i)$ are the surge attacks introduced from time t_{i_o} up to time $t_{i_o+L_a}$.

4.1.6. Geometric cyberattack

Geometric cyberattacks adhere to a strategy that gradually erodes the stability of the closed-loop system. It initiates with a gradual decay, which then accelerates exponentially as time progresses. This attack type attains its highest impact as the attack duration concludes. The initial move of the attacker involves introducing a constant value, labeled as β, to the genuine data (ensuring β remains considerably lower than the threshold value set within a min–max attack). In each subsequent time step, this initial deviation is magnified by a factor of $(1 + \alpha)$, where α falls within the range $(0, 1)$, until it reaches the maximum allowable attack magnitude. The two parameters, α and β, are prudently selected while accounting for the stability region, operational boundaries, and attack duration. Geometric attacks can be formulated as follows:

$$\bar{x}(t_i) = x(t_i) + \beta \times (1 + \alpha)^{i - i_o},\ \forall i \in [i_o, i_o + L_a] \tag{32a}$$

$$\bar{u}_{t2}(t_i) = u_{t2}(t_i) + \beta \times (1 + \alpha)^{i - i_o},\ \forall i \in [i_o, i_o + L_a] \tag{32b}$$

where the parameters α and β define the speed and magnitude of the geometric attack. $x(t_i)$ and $u_{t2}(t_i)$ are the true plant measurement and control input of the upper tier, respectively. L_a represents the length of the attack in terms of sampling periods. $\bar{x}(t_i)$ and $\bar{u}_{t2}(t_i)$ are the geometric attacks introduced from time t_{i_o} up to time $t_{i_o+L_a}$.

4.2. Machine-learning-based cyberattack detection

Utilizing a data-driven approach to construct the cyberattack detector offers numerous advantages. Firstly, given the potential access of attackers to process-behavior information, traditional first-principles model-based detection methods relying on predetermined statistical thresholds and false alarm biases become inadequate. Secondly, in real-world scenarios, the structure and parameters of the plant model are susceptible to alterations due to evolving operational conditions. In this context, adopting a data-centric approach for training the cyberattack detection mechanism proves resilient against both dynamic process changes and intricately crafted attacks.

In the realm of well-established machine learning approaches, neural networks (NN) have showcased their effectiveness in both supervised and unsupervised classification scenarios. In this particular study, we focus on a supervised classification task employing a two-class classification framework to determine whether a cyberattack has impacted the upper-tier control system.

When attacks involve data manipulation, they can manifest in various forms or patterns. Building a model to classify attack types can

lead to increased computational demands and model intricacy. Since our primary aim is to ascertain whether the upper-tier control has been subjected to an attack or not, we opt for a binary classification model. This approach simplifies the task and facilitates the identification of attack occurrences. Furthermore, to evaluate the effectiveness of the detector against attack patterns it has not encountered during training and validation, we introduce additional attack scenarios in the testing set that differ from those it has been exposed to previously.

The adopted neural network involves a sequence of nonlinear transformations, where neurons in the first hidden layer are computed from input data. Subsequent hidden neurons are derived from their preceding layer, culminating in the output being computed from neurons in the final hidden layer. These transformations occur in the form of activation functions involving biases and the weighted sum of inputs (or neurons from the previous layer). The fundamental structure of the utilized neural network model is depicted in Fig. 3, where each input corresponds to the feature-wise normalized control Lyapunov function computed from state measurements across a sequence of sampling instances. The control Lyapunov function captures the dynamics of all states of the system, making it an effective one-dimensional input feature for attack detection. While training the model, to make it generic, and to prevent overfitting, we adopted the standard practice of normalizing the training, testing, and validation datasets. Hence, while supplying the control Lyapunov function data during operation, this is normalized with respect to the mean and standard deviation of the training dataset, which is calculated prior to implementation of the detector in the process. This approach aids in aligning the data distributions and mitigates the influence of varying scales across features, thereby facilitating model training and enhancing model performance. The resulting output vector denotes the predicted class label, distinguishing between "cyberattack" and "no attack". The mathematical representation of the feed-forward neural network with two hidden layers can be formulated as:

$$\theta_j^{(1)} = g_1\left(\sum_{i=1}^{N_T} w_{ij}^{(1)}\hat{V}(x(t_i)) + b_j^{(1)}\right) \tag{33a}$$

$$\theta_j^{(2)} = g_2\left(\sum_{i=1}^{h_1} w_{ij}^{(2)}\theta_i^{(1)} + b_j^{(2)}\right) \tag{33b}$$

$$\theta_j^{(3)} = g_3\left(\sum_{i=1}^{h_2} w_{ij}^{(3)}\theta_i^{(2)} + b_j^{(3)}\right) \tag{33c}$$

$$\mathrm{y_{pred}} = \left[\theta_1^{(3)}, \theta_2^{(3)}, \ldots., \theta_H^{(3)}\right]^\top \tag{33d}$$

where $\theta_j^{(1)}$, $\theta_j^{(2)}$, and $\theta_j^{(3)}$ denote the output of the jth neuron of the first hidden layer, the second hidden layer, and the output layer, respectively. h_1 and h_2 stand for the neuron counts in the first and second hidden layers, while H signifies the number of class labels, equal to the number of neurons in the output layer. Within the input layer, the normalized control Lyapunov function of the complete state measurements at time t_i, denoted as $\hat{V}(x(t_i))$, serves as the input variable. The index $i = 1, \ldots, N_T$, with N_T being the duration of the time-varying trajectory for each input sample. The connections between neurons i and j in successive layers are weighted by $w_{ij}^{(k)}$, where $k = 1, 2, 3$. Additionally, the bias applied to the jth neuron in the kth layer is represented as $b_j^{(k)}$. Each layer receives input from its preceding layer and processes the input with optimized weights, biases, and nonlinear activation functions, represented by g_k. Within the output layer, the vector $\mathrm{y_{pred}}$ provides the probabilities for each class label concerning the analyzed sample. Notably, the neuron with the highest probability signifies the predicted class label.

The process of calculating training and testing accuracies entails computing the proportion of accurately classified samples relative to the total number of samples present within their respective training and testing datasets. In the development of a neural network model for cyberattack detection, closed-loop values of the control Lyapunov function are gathered over a fixed duration (N_T samples), encompassing various randomly initialized initial conditions. This is done both within and beyond the stability region Ω_ρ, ensuring coverage of a wide spectrum of allowable conditions. Given that $V(x)$ captures the dynamic characteristics of all states, it serves as an effective one-dimensional input feature for the attack detection problem. To improve training accuracy, an equivalent number of samples from each class are assembled. Each sample corresponds to a distinct set of initial conditions for the closed-loop system simulation. Further details of the model such as number of input neurons, activation functions, training, validation and testing accuracies are reported in Section 5.3.

Remark 3. To distinguish between dynamics in the control Lyapunov function caused by process fluctuations and cyberattacks, Gaussian-distributed noise is introduced into sensor signal measurements of the training and testing datasets. This accounts for both sensor noise and process disturbances, aiding the model to discern cyberattacks from fluctuations. In addition, a sliding window alarm is implemented, whereby the upper tier is deactivated only if the model identifies a cyberattack in three out of four consecutive sampling instances. This mechanism prevents accidental deactivation of the upper tier due to inherent process disturbances and ensures that, if inadvertently a cyberattack is detected at a single sampling instance due to process disturbances, the upper tier remains active. Such strategies are pivotal for the accurate differentiation of cyberattacks from process fluctuations.

5. Application to a chemical process

This section showcases the practical application of the suggested encrypted two-tier control framework in the context of a large-scale chemical process. We develop a nonlinear dynamical model based on first-principles modeling fundamentals. Subsequently, we employ it as the basis for constructing a first-principles-based encrypted LMPC. Alongside this, a set of encrypted PI controllers, capable of computing control input in an encrypted space, is formulated, and the control architecture is augmented with an ML-based cyberattack detector. Subsequently, we perform closed-loop simulations using the first-principles-based process model. Throughout these simulations, various cyberattacks are initiated, leading to the examination of multiple detection and control scenarios.

5.1. Process description and model development

The process considered is the synthesis of ethylbenzene (EB) through the conversion of ethylene (E) and benzene (B). The primary reaction, termed as "primary", is characterized as a second-order, exothermic, and irreversible reaction, in conjunction with two supplementary side reactions. These reactions occur within two non-isothermal, well-mixed continuous stirred tank reactors (CSTRs). The chemical reactions taking place are articulated as follows:

$$\mathrm{C_2H_4 + C_6H_6 \rightarrow C_8H_{10}} \quad \text{(primary)} \tag{34a}$$

$$\mathrm{C_2H_4 + C_8H_{10} \rightarrow C_{10}H_{14}} \tag{34b}$$

$$\mathrm{C_6H_6 + C_{10}H_{14} \rightarrow 2C_8H_{10}} \tag{34c}$$

The state variables are the concentration of ethylene, benzene, ethylbenzene, di-ethylbenzene, and the reactor temperature for each CSTR_i, $i = (1, 2)$, in deviation terms, that is: $x^\top = [C_{E_1} - C_{E_{1s}},\ C_{B_1} - C_{B_{1s}},\ C_{EB_1} - C_{EB_{1s}},\ C_{DEB_1} - C_{DEB_{1s}},\ T_1 - T_{1s},\ C_{E_2} - C_{E_{2s}},\ C_{B_2} - C_{B_{2s}},\ C_{EB_2} - C_{EB_{2s}},\ C_{DEB_2} - C_{DEB_{2s}},\ T_2 - T_{2s}]$. The subscript "$s$" denotes the steady-state value. The rate of heat removal for the two reactors $[Q_1 - Q_{1s},\ Q_2 - Q_{2s}]$ are the control inputs manipulated by the lower tier using encrypted PI controllers, which are bounded by the closed sets, $[-10^4\ \mathrm{kW},\ 2 \times 10^3\ \mathrm{kW}]$ and $[-1.5 \times 10^4\ \mathrm{kW},\ 5 \times 10^3\ \mathrm{kW}]$ respectively. The inlet feed concentrations for each reactor, $[C_{E_{o1}} - C_{E_{o1s}},\ C_{B_{o1}} - C_{B_{o1s}},\ C_{E_{o2}} - C_{E_{o2s}},\ C_{B_{o2}} - C_{B_{o2s}}]$, are the control inputs manipulated

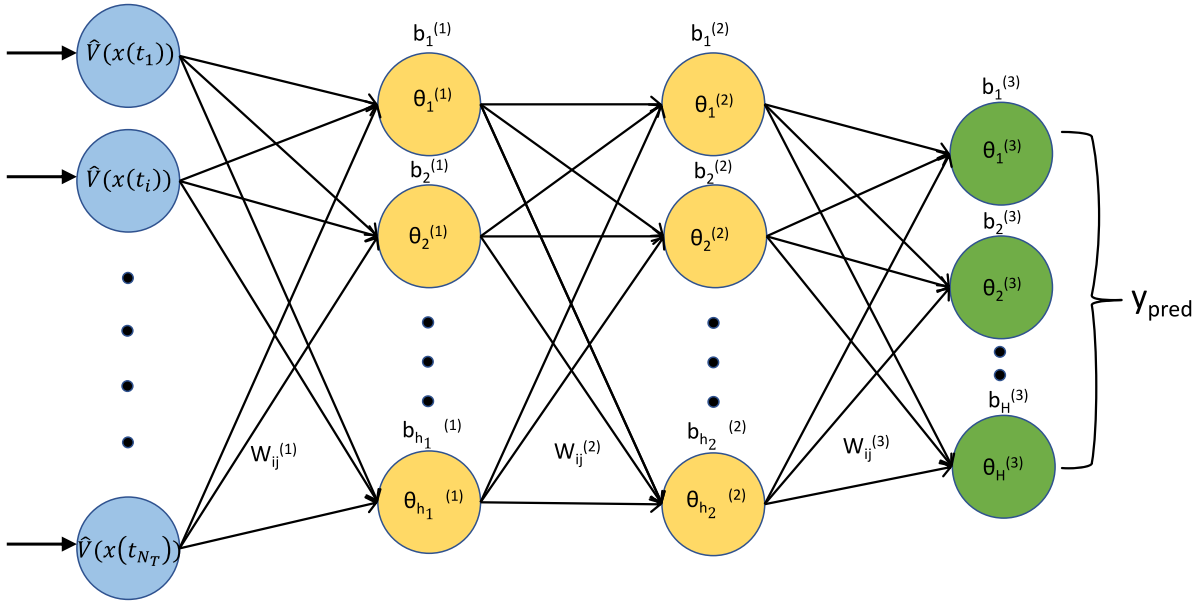

Fig. 3. Feed-forward neural network structure of the proposed ML-based cyberattack detector.

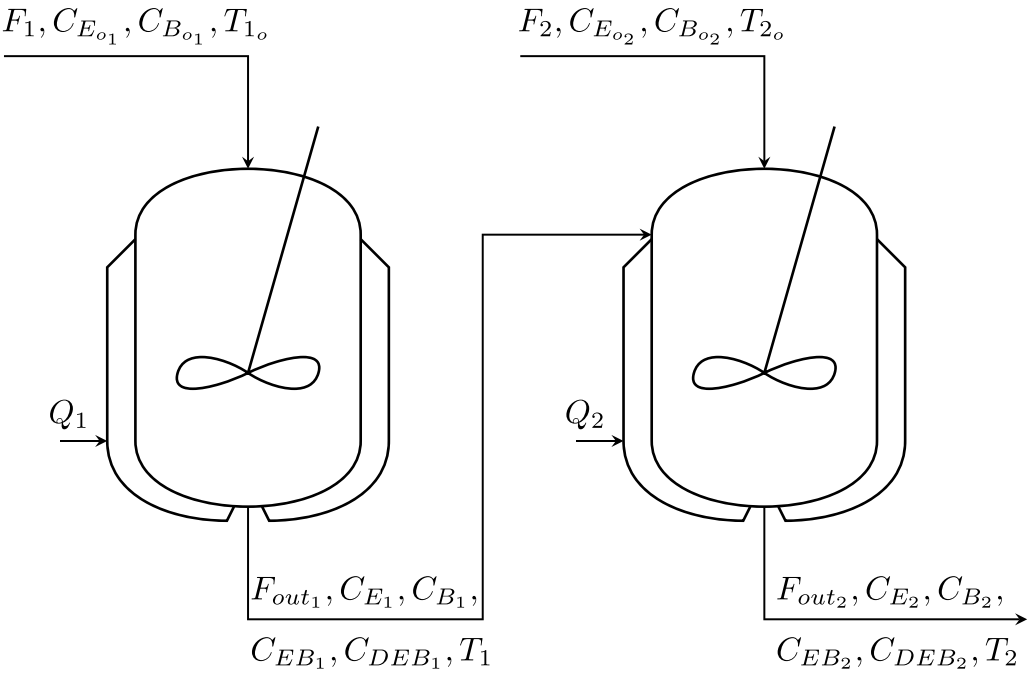

Fig. 4. Process schematic featuring two CSTRs connected in series.

by the upper tier using an encrypted MPC, which are bounded by the closed sets $[-2.5\,\text{kmol/m}^3,\ 2.5\,\text{kmol/m}^3]$, $[-2.5\,\text{kmol/m}^3,\ 2.5\,\text{kmol/m}^3]$, $[-3\,\text{kmol/m}^3,\ 3\,\text{kmol/m}^3]$, and $[-3\,\text{kmol/m}^3,\ 3\,\text{kmol/m}^3]$, respectively. The control objective is to maintain the operation of both CSTRs at their unstable equilibrium state through the utilization of the encrypted two-tier control scheme, employing quantized states and inputs for computation and actuation. Through the application of mass and energy balance principles, the foundational model for the CSTRs is constructed. An illustrative visualization of this model is presented in Fig. 4. In particular, the dynamic representation of the initial CSTR is captured by the subsequent set of ordinary differential equations (ODEs):

$$\frac{\mathrm{d}C_{E_1}}{\mathrm{d}t} = \frac{F_1 C_{Eo_1} - F_{out_1} C_{E_1}}{V_1} - r_{1,1} - r_{1,2} \tag{35a}$$

$$\frac{\mathrm{d}C_{B_1}}{\mathrm{d}t} = \frac{F_1 C_{Bo_1} - F_{out_1} C_{B_1}}{V_1} - r_{1,1} - r_{1,3} \tag{35b}$$

$$\frac{\mathrm{d}C_{EB_1}}{\mathrm{d}t} = \frac{-F_{out_1} C_{EB_1}}{V_1} + r_{1,1} - r_{1,2} + 2r_{1,3} \tag{35c}$$

$$\frac{\mathrm{d}C_{DEB_1}}{\mathrm{d}t} = \frac{-F_{out_1} C_{DEB_1}}{V_1} + r_{1,2} - r_{1,3} \tag{35d}$$

$$\frac{\mathrm{d}T_1}{\mathrm{d}t} = \frac{T_{1_o} F_1 - T_1 F_{out_1}}{V_1} + \sum_{j=1}^{3} \frac{-\Delta H_j}{\rho_1 C_p} r_{1,j} + \frac{Q_1}{\rho_1 C_p V_1} \tag{35e}$$

The dynamic model of the second CSTR is represented by the following ODEs:

$$\frac{\mathrm{d}C_{E_2}}{\mathrm{d}t} = \frac{F_2 C_{Eo_2} + F_{out_1} C_{E_1} - F_{out_2} C_{E_2}}{V_2} - r_{2,1} - r_{2,2} \tag{36a}$$

Table 1
Parameter values and steady-state values of the first-principles-based dynamic model.

$T_{1o} = T_{2o} = 350\,\text{K}$	$T_{1s} = 321.15\,\text{K}$
$V_1 = V_2 = 60\,\text{m}^3$	$T_{2s} = 442.99\,\text{K}$
$F_1 = 43.2\,\text{m}^3/\text{h}$	$F_2 = 47.87\,\text{m}^3/\text{h}$
$F_{out_1} = F_1$	$F_{out_2} = F_1 + F_2$
$C_{E_{o_1}} = 4.43\,\text{kmol/m}^3$	$C_{E_{1_s}} = 4.33\,\text{kmol/m}^3$
$C_{B_{o_1}} = 5.54\,\text{kmol/m}^3$	$C_{B_{1_s}} = 5.55\,\text{kmol/m}^3$
$C_{E_{o_2}} = 4.02\,\text{kmol/m}^3$	$C_{E_{2_s}} = 0.196\,\text{kmol/m}^3$
$C_{B_{o_2}} = 5.02\,\text{kmol/m}^3$	$C_{B_{2_s}} = 1.31\,\text{kmol/m}^3$
$C_{EB_{1_s}} = 0.53\,\text{kmol/m}^3$	$C_{EB_{2_s}} = 4.22\,\text{kmol/m}^3$
$C_{DEB_{1_s}} = 8.76 \times 10^{-4}\,\text{kmol/m}^3$	$C_{DEB_{2_s}} = 0.0078\,\text{kmol/m}^3$
$k_1 = 1.528 \times 10^6\,\text{m}^3\,\text{kmol}^{-1}\,\text{s}^{-1}$	$E_1 = 71\,160\,\text{kJ/kmol}$
$k_2 = 2.778 \times 10^4\,\text{m}^3\,\text{kmol}^{-1}\,\text{s}^{-1}$	$E_2 = 83\,680\,\text{kJ/kmol}$
$k_3 = 0.4167\,\text{m}^3\,\text{kmol}^{-1}\,\text{s}^{-1}$	$E_3 = 62\,760\,\text{kJ/kmol}$
$\rho_1 = 639.153\,\text{kg/m}^3$	$\rho_2 = 607.504\,\text{kg/m}^3$
$\Delta H_1 = -1.04 \times 10^5\,\text{kJ/kmol}$	$\Delta H_2 = -1.02 \times 10^5\,\text{kJ/kmol}$
$\Delta H_3 = -5.5 \times 10^2\,\text{kJ/kmol}$	$C_p = 2.411\,\text{kJ}\,\text{kg}^{-1}\,\text{K}^{-1}$
$Q_{1_s} = -1074.63\,\text{kW}$	$Q_{2_s} = -6768.83\,\text{kW}$
$R = 8.314\,\text{kJ}\,\text{kmol}^{-1}\,\text{K}^{-1}$	

$$\frac{dC_{B_2}}{dt} = \frac{F_2 C_{Bo_2} + F_{out_1} C_{B_1} - F_{out_2} C_{B_2}}{V_2} - r_{2,1} - r_{2,3} \tag{36b}$$

$$\frac{dC_{EB_2}}{dt} = \frac{F_{out_1} C_{EB_1} - F_{out_2} C_{EB_2}}{V_2} + r_{2,1} - r_{2,2} + 2r_{2,3} \tag{36c}$$

$$\frac{dC_{DEB_2}}{dt} = \frac{F_{out_1} C_{DEB_1} - F_{out_2} C_{DEB_2}}{V_2} + r_{2,2} - r_{2,3} \tag{36d}$$

$$\frac{dT_2}{dt} = \frac{T_{2_o} F_2 + T_1 F_{out_1} - T_2 F_{out_2}}{V_2} + \sum_{j=1}^{3} \frac{-\Delta H_j}{\rho_2 C_p} r_{2,j} + \frac{Q_2}{\rho_2 C_p V_2} \tag{36e}$$

where the reaction rates are calculated by the following expressions:

$$r_{i,1} = k_1 e^{\frac{-E_1}{RT_i}} C_{E_i} C_{B_i} \tag{37a}$$

$$r_{i,2} = k_2 e^{\frac{-E_2}{RT_i}} C_{E_i} C_{EB_i} \qquad i = 1, 2 \text{ (reactor index)} \tag{37b}$$

$$r_{i,3} = k_3 e^{\frac{-E_3}{RT_i}} C_{DEB_i} C_{B_i} \tag{37c}$$

The process model parameter values and the corresponding steady-state values are given in Table 1.

5.2. Implementing encryption in the two-tier control architecture

Prior to integrating encryption–decryption into a process, the selection of parameters, namely d, l_1, and l_2 is performed. An evaluation of the extreme feasible states and inputs guides the derivation of the integer bit count, denoted as $l_1 - d$. The upper limit in the $\mathbb{Q}_{l_1,d}$ set is obtained via the formula $2^{l_1-d-1} - 2^{-d}$, whereas the lower limit is -2^{l_1-d-1}. The choice of the quantization parameter d rests on the desired accuracy and range of state and input values. Additionally, l_2 is chosen to exceed l_1. In alignment with this methodology, for the CSTR studied in this section, $l_1 - d$ is calculated to be 16, from which l_1 and d are then fixed. Within the set $\mathbb{Q}_{l_1,d}$, rational numbers are separated by a resolution of 2^{-d}, indicating that higher d values result in lower quantization errors. For simulation purposes, we opt for $d = 8$ as it yields nearly identical closed-loop state trajectories in comparison to the unencrypted case (Suryavanshi et al., 2023; Kadakia et al., 2023). These cited works also illustrate how the choice of the quantization parameter impacts closed-loop performance and stability across different values of d. For $d = 8$, we obtain $l_1 = 24$ and, since it is imperative that $l_2 > l_1$ for the subsequent bijective mapping, l_2 is selected as 30. With the quantization parameters defined, the next step entails the quantization of states and inputs, followed by their encryption via the Paillier Encryption algorithm. The implementation of the Paillier Encryption procedure is done through Python's "phe" module, PythonPaillier (Data61, 2013).

Remark 4. As mentioned earlier, the implementation of encryption requires quantization of real-number valued signals to a fixed dataset denoted as $\mathbb{Q}_{l_1,d}$. The selection of quantization parameter $d = 8$ is justified by its enhanced control performance in comparison to lower values of d. The time needed for encryption computation can be divided into five distinct components, the time spent for: quantization of real data ($g_{l_1,d}$), bijective mapping ($f_{l_2,d}$), encryption, decryption, and inverse mapping ($f^{-1}_{l_2,d}$). Kadakia et al. (2023) underscored that the encryption phase, followed by decryption, accounts for the majority of the time spent. Moreover, the time remains unaffected by the chosen quantization parameter. The three remaining mathematical operations contribute only a minimal portion of the total time spent on encryption–decryption. While the time taken for these operations does increase with quantization, the increment is insignificant compared to the total time spent, and the advantage of improved control performance using a higher quantization parameter greatly outweighs the slight increase in time. Hence, a quantization parameter of $d = 8$ is adopted across all cases where encryption is implemented in this study.

5.2.1. Implementation of the encrypted lower-tier control system

In the lower tier, control input computations are confined to linear mathematical operations, ensuring their execution within an encrypted space that guarantees cyber-security. The selection of lower-tier controlled inputs, which possess the capability to stabilize the entire system, is a pivotal task that requires adherence to a well-defined procedure. The procedure includes linearization of the nonlinear dynamical model about its operating steady-state, yielding a 10-dimensional state space model mirroring the number of states, governed by two control inputs—the heat removal rate for each CSTR. A, B, C, and D matrices were created for the state–space model $\dot{x} = Ax + Bu$ and $y = Cx + Du$, where y are the observed measurements from the system. Subsequently, leveraging the Cohen–Coon tuning method, the control input gains are calibrated and further refined through multiple simulations conducted on the nonlinear dynamical model. Subsequently, the integral terms are omitted, substituting only proportional terms, $u = Kx$ in the state–space model, resulting in $\dot{x} = Ax + BKx$. The eigenvalues of $(A + BK)$ are then computed and verified to exhibit negative real components. This ensures asymptotic stability for the controllers when applied to the linearized model over the operating steady state. The inclusion of the integral term serves to eradicate offsets, thereby contributing to the refinement of closed-loop performance. Although excluded in the eigenvalue computations, the integral terms were meticulously adjusted through a series of simulations using the nonlinear dynamical model. Next, via extensive simulations of the nonlinear system under the lower-tier controller, the controller is verified to adhere to the criteria outlined in Eq. (12), confirming that it can exponentially stabilize the system within the two-tier encrypted control framework.

5.2.2. Implementation of the encrypted LMPC in the upper-tier control system

The first-principles model, expressed by Eq. (35), serves as the foundational process model within the LMPC framework. For solving the constrained nonlinear, non-convex optimization problem, we leverage the Python module of the IPOPT software (Wächter and Biegler, 2006). Consequently, the resultant solution is a local optimum, not a global one. This is a limitation due to the nature of the optimization that a global optimum cannot be found for such a problem (Bomze et al., 2010). The process of solving this optimization problem involves defining constraints for the LMPC. IPOPT constructs a feasibility region and employs an iterative methodology to progressively navigate towards the optimal solution by traversing the interior of the feasibility region. This approach incorporates two key parameters: the maximum number of iterations and a validation error. These parameters function as the stopping criteria within the optimization problem. If either of these conditions is met, IPOPT employs the final computed value as the solution for the given instance. Conversely, if neither of these criteria

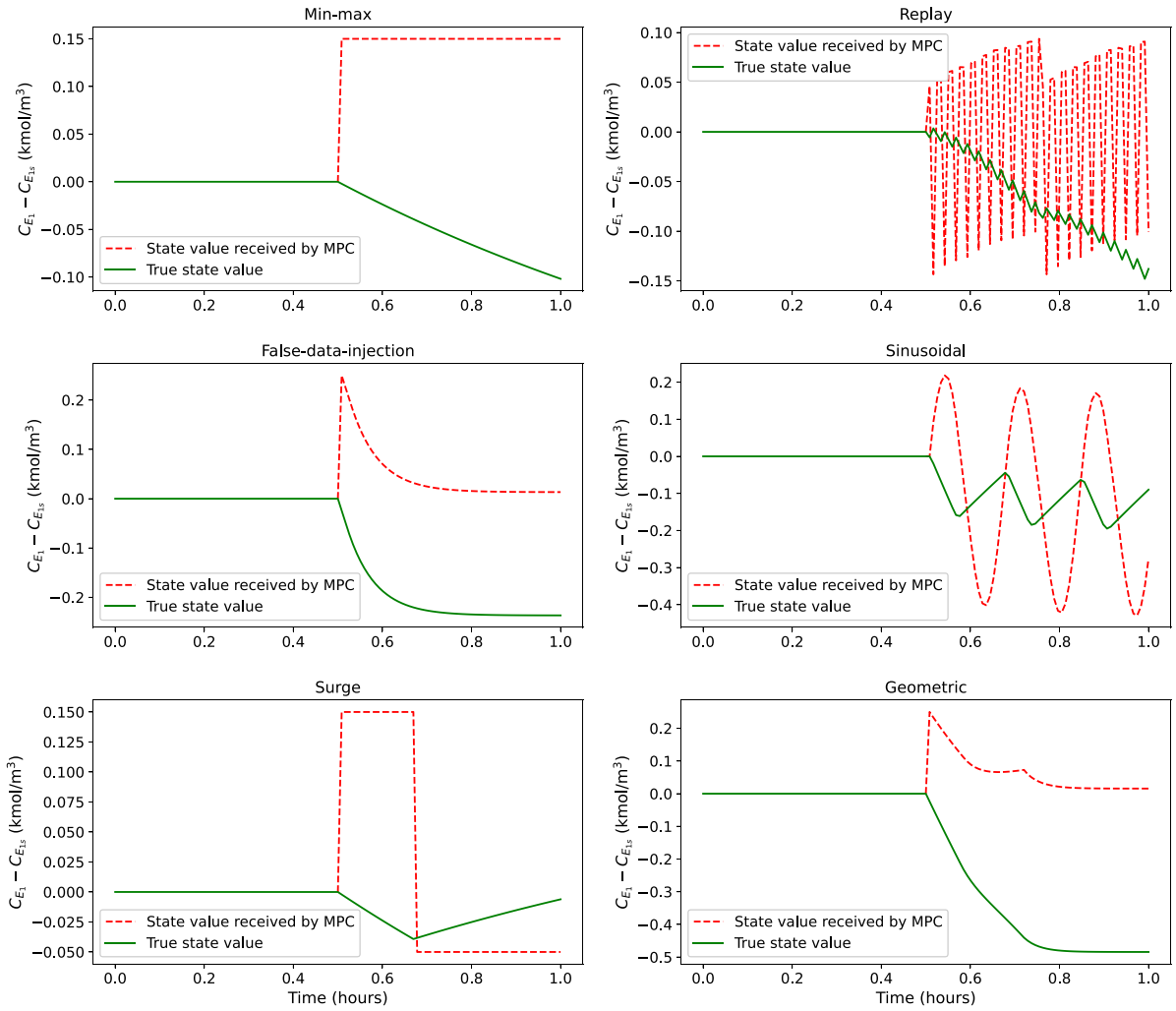

Fig. 5. True state value of $C_{E_1} - C_{E_{1s}}$ (green solid line) and state value of $C_{E_1} - C_{E_{1s}}$ received by the MPC (red dashed line) for all the cyberattacks discussed.

is satisfied, IPOPT reports the suboptimal values calculated in the last iteration, but the LMPC utilizes the control input calculated by the backup controller.

To assess the cost function of the LMPC over the prediction horizon, the integration step h_c is determined as $10^{-2} \times \Delta$ using the first-principles model. The positive definite matrix P in the control Lyapunov function $V = x^\top Px$ is selected as diag [250 500 500 1000 0.3 250 250 500 1000 0.6], drawing from extensive simulations. The LMPC framework employs a prediction horizon of $N = 2$. The stability criterion is defined as $\rho = 100$. Additionally, the criterion $\rho_{\min} = 1$ is the smaller level set of the Lyapunov function where the state is desired to be trapped. The weight matrices Q_1 and Q_2 in the LMPC cost function are chosen as $Q_1 = \text{diag}\,[2000\ 2000\ 5000\ 5\ 5\ 2000\ 2000\ 5000\ 2\ 2]$ and $Q_2 = \text{diag}\,[1\ 1\ 6\ 8]$, respectively. The cost function is defined as $L(x, u_{t2}) = x^\top Q_1 x + u_{t2}^\top Q_2 u_{t2}$.

5.2.3. Sampling time criteria with encryption

To implement encryption within a practical context, it is essential to ensure that the sampling time, Δ, surpasses the combined maximum duration required for encryption and decryption of all states and control inputs. Furthermore, it should accommodate the maximum time necessary for computing control actions at each sampling instance across the considered quantization parameter (d). This condition holds true for both the upper- and lower-tier control systems within an encrypted two-tier control framework. Mathematically,

$$\Delta_i > \max\,(\text{Encryption–decryption time})_i + \max\,(\text{Control input computation time})_i \tag{38}$$

where $i = \{1, 2\}$, with $i = 1$ and $i = 2$ representing the lower and upper control tier, respectively. In the discussed example, the sampling time Δ is chosen as 30 s. This decision is made while taking into account the previously mentioned requirement to implement the encryption process. Eq. (38) does not include the communication time required for signal transmission. This is because the two-tier encrypted control architecture, discussed within the context of SCADA systems, relies on networked communication, which is extremely efficient and rapid. However, networked communication also exposes the system to cyberattacks, which is a vulnerability that we aim to mitigate in this research by introducing encryption to these communication channels.

Remark 5. In the context of the discussed two-tier encrypted control architecture, the lower and upper tiers operate independently, maintaining distinct public keys for encryption and private keys for decryption. Consequently, they possess the flexibility to adopt different sampling times. For the CSTR example studied in this work, both tiers maintain identical sampling times. If certain control inputs necessitate shorter sampling periods and more frequent actuation, it is advisable to allocate them to the lower tier. The lower tier is a set of linear controllers and, hence, can compute control inputs more rapidly than an advanced nonlinear control scheme such as MPC employed in the upper tier. Also, as the lower tier does not perform encryption and decryption within the networked communication channels, it has less stringent sampling time constraints. Furthermore, strategies employing

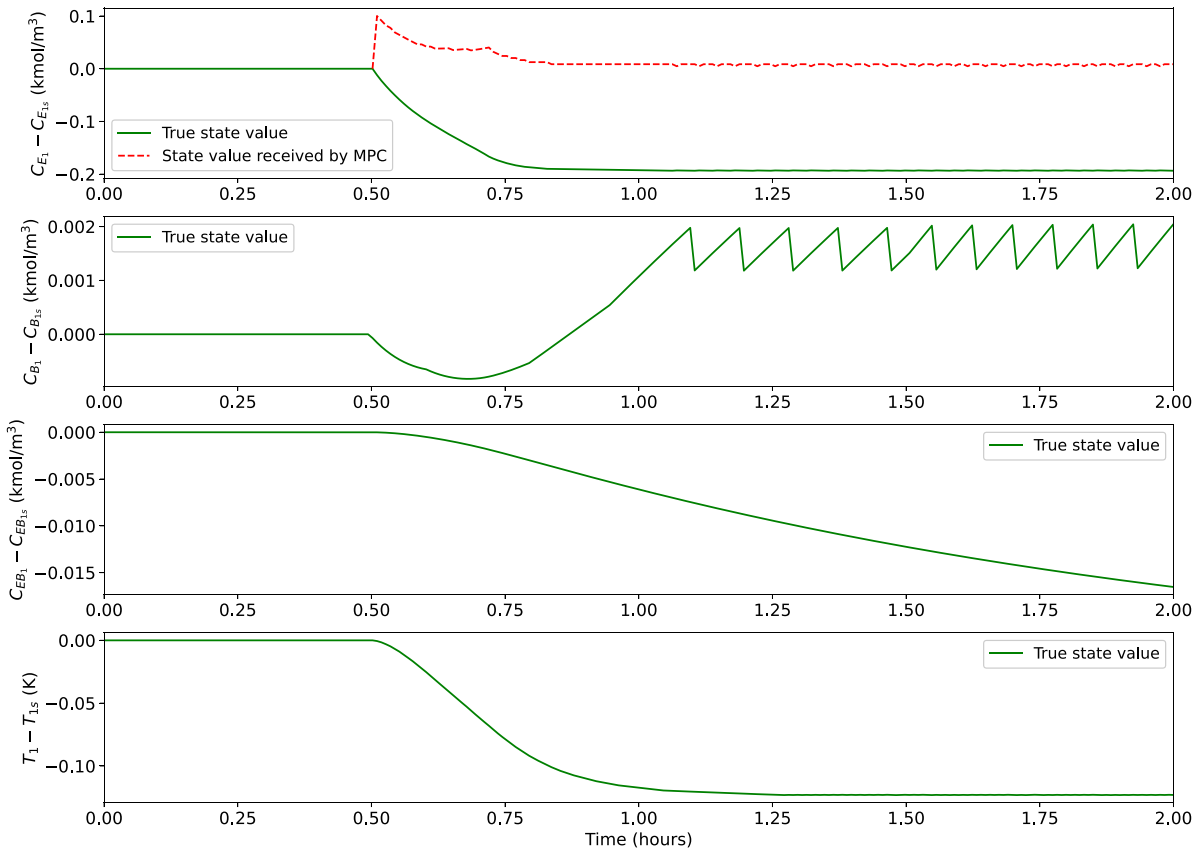

Fig. 6. True state values (green solid line) and state value received by the MPC (red dashed line) for CSTR 1 without detection and reconfiguration mechanisms when a geometric cyberattack is launched at $t = 0.5$ h.

two-tier control to address challenges posed by delayed and asynchronous signals have been demonstrated in prior studies of Liu et al. (2008, 2010). For systems incorporating delayed and asynchronous signals, these signals can be transmitted to the upper tier while applying control inputs through a sample-and-hold procedure. However, the primary motivation behind the adoption of the two-tier design in this research is the cyber-vulnerability of the upper tier due to the need to compute nonlinear control inputs without the safeguard of an encrypted computational environment.

5.3. Cyberattack detector training and testing

In the upper-tier control system, the cyberattacks take the form of data manipulation. The objective involves crafting a detector capable of recognizing cyberattacks based on familiar data manipulation patterns as well as those it has not encountered previously. To accomplish this, a feed-forward neural network (FNN) is used to identify cyberattacks. The FNN is trained with min–max, replay, sinusoidal, and false-data-injection attacks. The FNN underwent testing with the aforementioned attacks, along with the inclusion of surge and geometric attacks. The outcome of the FNN is categorized into two classes: "cyberattack" and "no attack". Each data point in the dataset represents a 1×40 array of $V(x)$ values. To instill variability, we employed a spectrum of initial conditions, mirroring a range of process scenarios. The activation of an attack was randomly timed between $i_o \in [5, 35]$ to create diverse durations and occurrences during system operation. Throughout the training phase, a randomized approach was adopted, wherein an attack would be simulated on a single state measurement for each CSTR at random intervals. In the testing phase, a similar random approach was followed, wherein cyberattacks were introduced on either one or multiple state measurements or control inputs.

To build the training and validation set, we conducted extensive closed-loop simulations, resulting in a dataset comprising 6000 data points. Each class ("cyberattack" and "no attack") contained 3000 data points. For the cyberattack class, 750 data points per attack type were included in the training. The dataset was divided into an 80:20 ratio for training and validation purposes. Employing feature-wise normalization prevented overfitting and enhanced results. For the testing phase, a separate set of 1200 data points was generated — 600 for each class and 100 data points for each cyberattack type. To account for real-world process fluctuations and avoid mistaking minor variations as cyberattacks, bounded Gaussian white noise was incorporated into each sensor measurement, for all the data points. By bounding the noise, the tail ends of the Gaussian distributed noise are eliminated before being applied. The cited work of Singh et al. (2023) proposes methods to deal with tail-ends in Gaussian-distributed noise. The sensor noises were constrained within the following bounds: $|\omega_i| \leq 0.1, \forall i = \{1, 2, 3, 6, 7, 8\}$; $|\omega_i| \leq 0.0003, \forall i = \{4, 9\}$; $|\omega_i| \leq 0.35, \forall i = \{5, 10\}$; these Gaussian noise distributions have zero mean and standard deviations $|\sigma_i| \leq 0.03, \forall i = \{1, 2, 3, 6, 7, 8\}$; $|\sigma_i| \leq 0.0001, \forall i = \{4, 9\}$; $|\sigma_i| \leq 0.1, \forall i = \{5, 10\}$. In this context, the subscripts are associated with different system states. Subscripts 1, 2, 3, 4, and 5 denote the concentrations of ethylene, benzene, ethylbenzene, di-ethyl benzene, and reactor temperature for CSTR 1, respectively. Similarly, subscripts 6, 7, 8, 9, and 10 correspond to the concentrations of ethylene, benzene, ethylbenzene, di-ethyl benzene, and reactor temperature for CSTR 2, respectively.

The design of the feed-forward neural network structure followed a systematic approach. It comprised 40 input neurons, each corresponding to normalized control Lyapunov function values derived from the previous 40 sampling instances. The FNN was designed with two hidden layers, while the output neurons were set to 2, aligning with the binary classification task at hand. Fixing the number of neurons in the

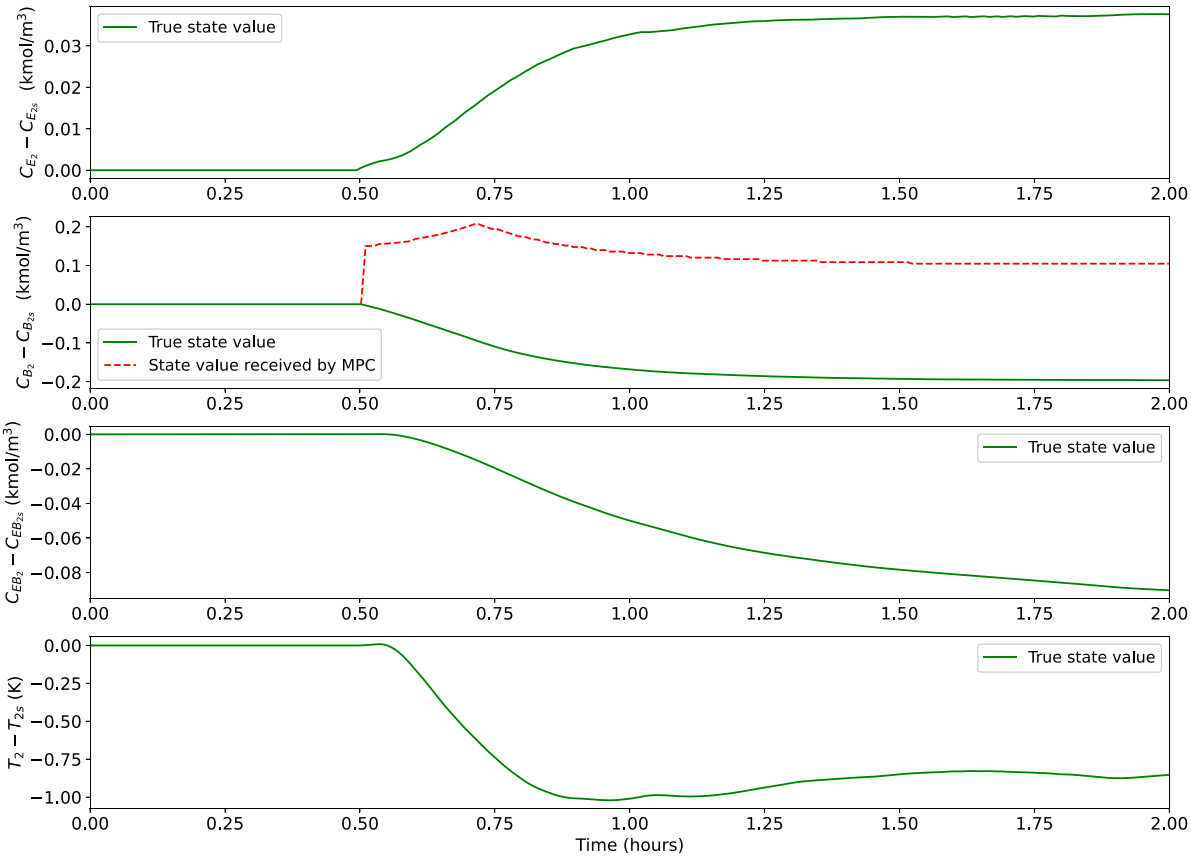

Fig. 7. True state values (green solid line) and state value received by the MPC (red dashed line) for CSTR 2 without detection and reconfiguration mechanisms when a geometric cyberattack is launched at $t = 0.5$ h.

hidden layers, selecting the optimizer, and specifying activation functions before the hidden layers was established through a meticulous grid search process. The number of epochs was fixed to 100 during the grid search. The objective was to identify the optimal combination of hyperparameters that yielded the lowest validation loss. Based on the results, the configuration of the neural network architecture included 60 neurons in the first hidden layer and 25 neurons in the second hidden layer. To mitigate the risk of overfitting, a dropout ratio of 0.2 was applied after each hidden layer. The activation functions employed were as follows: hyperbolic tangent after the input layer, rectified linear unit (ReLU) after the first hidden layer, and softmax after the second hidden layer. Upon tuning with these hyperparameters, the model underwent 1000 epochs of training using the Adam optimizer with the objective to minimize the sparse categorical cross-entropy loss. Throughout the training, emphasis was placed on conserving the model configuration that exhibited the lowest validation loss. This meticulous approach facilitated the development of an effective and well-optimized neural network model for subsequent testing and evaluation. The training, validation, and testing accuracies for the model are 99.87%, 99.92%, and 99.83, % respectively.

Remark 6. As outlined in Section 5.3, the cyberattack is introduced randomly within the sampling instances ranging from [5, 35], covering a span of 40 instances from which data is gathered for the control Lyapunov function for a single data point. Attacks launched after sampling instance 30 pose a relatively higher challenge for cyberattack detection algorithms. Within the following sampling instances, these attacks may not induce substantial deviations in the process dynamics. This is due to the model being trained with noisy data to prevent ordinary process fluctuations from being misidentified as cyberattacks. However, as the attacks persist and gradually push the system away from the desired stability region $\Omega_{\rho_{\min}}$ (but still within Ω_{ρ}), their detectability becomes more feasible. Hence, while the accuracies might not reach 100%, practical implementation within a system reveals the potential to achieve cyberattack detection with near-perfect accuracy and very slightly extended response times.

Remark 7. Di-ethyl benzene is an unintended byproduct that emerges within the reaction scheme elucidated in Eq. (34). It exists in minimal quantities within both CSTRs and is not a direct control input. Consequently, in the process of randomly initiating cyberattacks on the state values received by the MPC for training, validation, and testing datasets, no cyberattacks are launched on the state values of di-ethyl benzene. This omission stems from the recognition that cyberattacks on state values of di-ethyl benzene would exert no discernible influence on the overall process dynamics. For this reason, cyberattacks are exclusively aimed at the eight other state values received by the MPC, as well as all four control inputs computed by the MPC in the upper-tier control system. Given its trace presence, visual depictions of its concentration are not included in this section. However, di-ethyl benzene is considered as a system state for the purpose of process modeling and MPC calculations. Consequently, all results presented account for its presence within the system.

5.4. Two-tier control architecture without cyberattack detection and reconfiguration mechanisms

In this section, we illustrate the two-tier control architecture without incorporating any detection and control reconfiguration mechanism. Fig. 5 visually illustrates all six discussed cyberattacks. The cyberattacks are launched at time $t = 0.5$ h. The true state measurements of the concentration of ethylene in CSTR 1 of the process

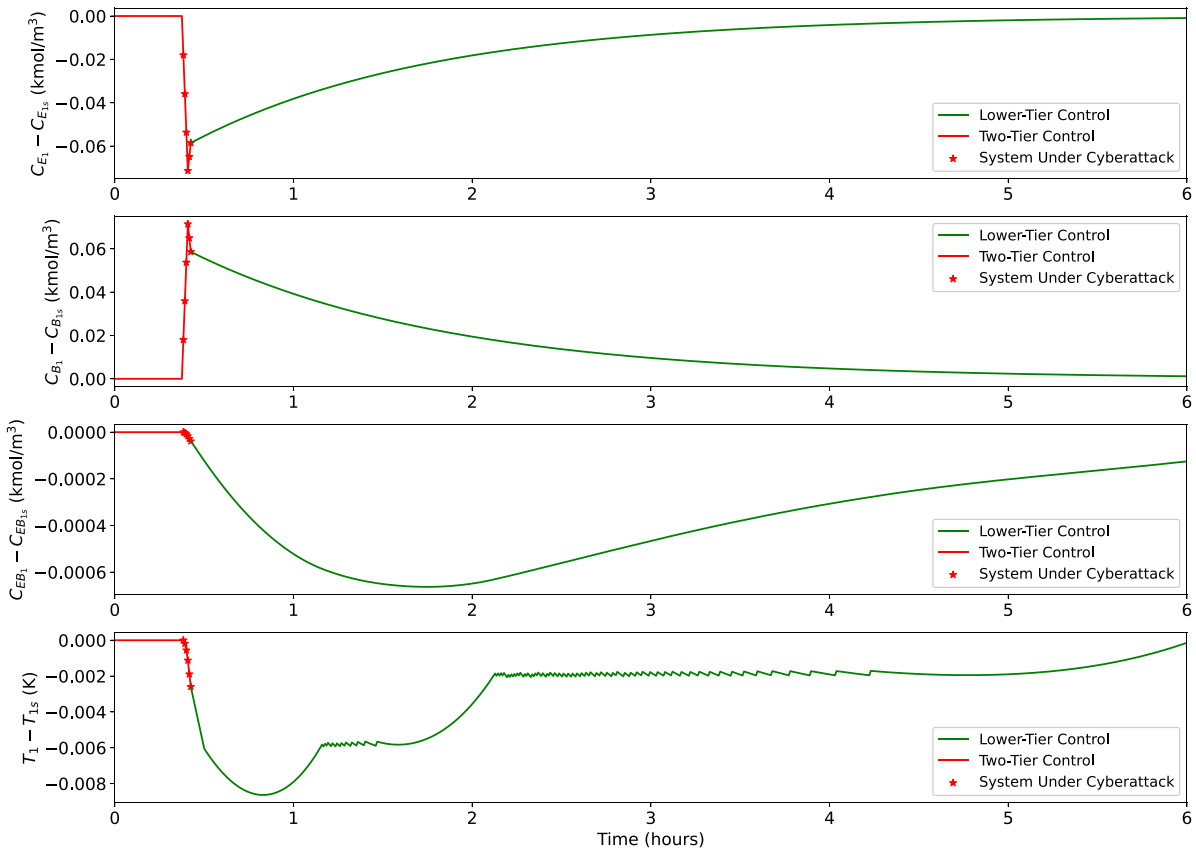

Fig. 8. State profiles of CSTR 1 under encrypted two-tier control (red line), encrypted two-tier control under cyberattack (red line with stars), and encrypted lower tier post-reconfiguration (green line), when a surge attack is launched on the upper-tier control inputs at $t = 23$ min.

network, depicted by the solid green line, stand in contrast to the manipulated state values received by the MPC during a cyberattack. The altered values, indicated by the dashed red line, emerge due to the manipulation of the true state values received by the MPC during the cyberattack. Evidently, the actual state values and the received values by the MPC diverge in opposite directions as the actuation is executed based on the received values, rather than the authentic state values.

To portray the overall impact of a cyberattack on the system in the absence of a detection mechanism, a geometric cyberattack is executed on two state measurements of the upper tier at time $t = 0.5$ h. The attack targets the state values associated with the concentration of ethylene in CSTR 1 and the concentration of benzene in CSTR 2 that are received by the MPC. As evident from Figs. 6 and 7, the cyberattack does not destabilize the system beyond the stability region Ω_ρ. The final value of the control Lyapunov function $V(x)$ at $t = 2$ h is 33.65 which is within the stability limit, $\rho = 100$. Nevertheless, it does lead to a continued reduction in the concentration of ethyl benzene in CSTR 2—the desired product—resulting in economic loss. The lower tier, responsible for controlling the heat inputs to both CSTRs and is fully safeguarded against cyberattacks, prevents attacks on the upper tier from completely destabilizing the system. However, the integration of a machine learning-based cyberattack detection mechanism can deactivate the upper tier, thereby ensuring system stabilization within the desired stability region $\Omega_{\rho_{\min}}$. Furthermore, conventional detection mechanisms based on fail-safe boundary conditions, like identifying an attack when the value of the control Lyapunov function surpasses $\rho = 100$, would prove inadequate in detecting an intelligent cyberattack.

5.5. Simulation results of the encrypted two-tier control architecture with cyberattack detection and re-configuration mechanisms

In this section, we employ the encrypted two-tier control architecture, featuring a machine learning-based cyberattack detector and a reconfiguration mechanism to disable the upper tier upon cyberattack detection. Two distinct scenarios are presented: one where the system operates at an unstable steady-state and the other where the system is converging to an unstable steady-state while remaining within the stability region Ω_ρ. The objective of intelligent cyberattacks is to inflict harm on the process yield without causing the system to exit the stability region. As a result, we do not delve into cyberattacks launched when the system states are outside Ω_ρ, as conventional detection mechanisms are sufficient for addressing such cases.

In both scenarios under consideration, the cyberattack is initiated at $t = 0.383$ h or 23 min of process time. As mentioned, the upper-tier control inputs are the inlet concentration of ethylene and benzene for each CSTR and the lower-tier control inputs are the heat removal rates for each CSTR. Figs. 8–10 depict the first scenario, where the control inputs computed by the MPC before encryption are manipulated via a surge attack when the system is operating at its unstable steady-state. Figs. 11–13 depict the second scenario, where the state values of the system received by the MPC after decryption are manipulated via a geometric attack when the system is converging to its unstable steady state. In all the figures in Section 5.5, the operating control scheme is illustrated through different colored lines. The red line depicts the system under the two-tier encrypted control scheme, the red line marked with stars depicts the system under the two-tier encrypted control scheme during a cyberattack, and the green line depicts the system under solely the lower-tier control scheme after the cyberattack has been detected, and the system has been reconfigured. It is worth noting that the ML-based cyberattack detector was not trained on the geometric and surge attack patterns. Yet, the detector demonstrated its ability to promptly identify these attacks.

In Figs. 8 to 10, during the initial 23 min of the process time, flat trajectories are observed for all the states and control inputs as the system is operating at its unstable steady-state under the two-tier

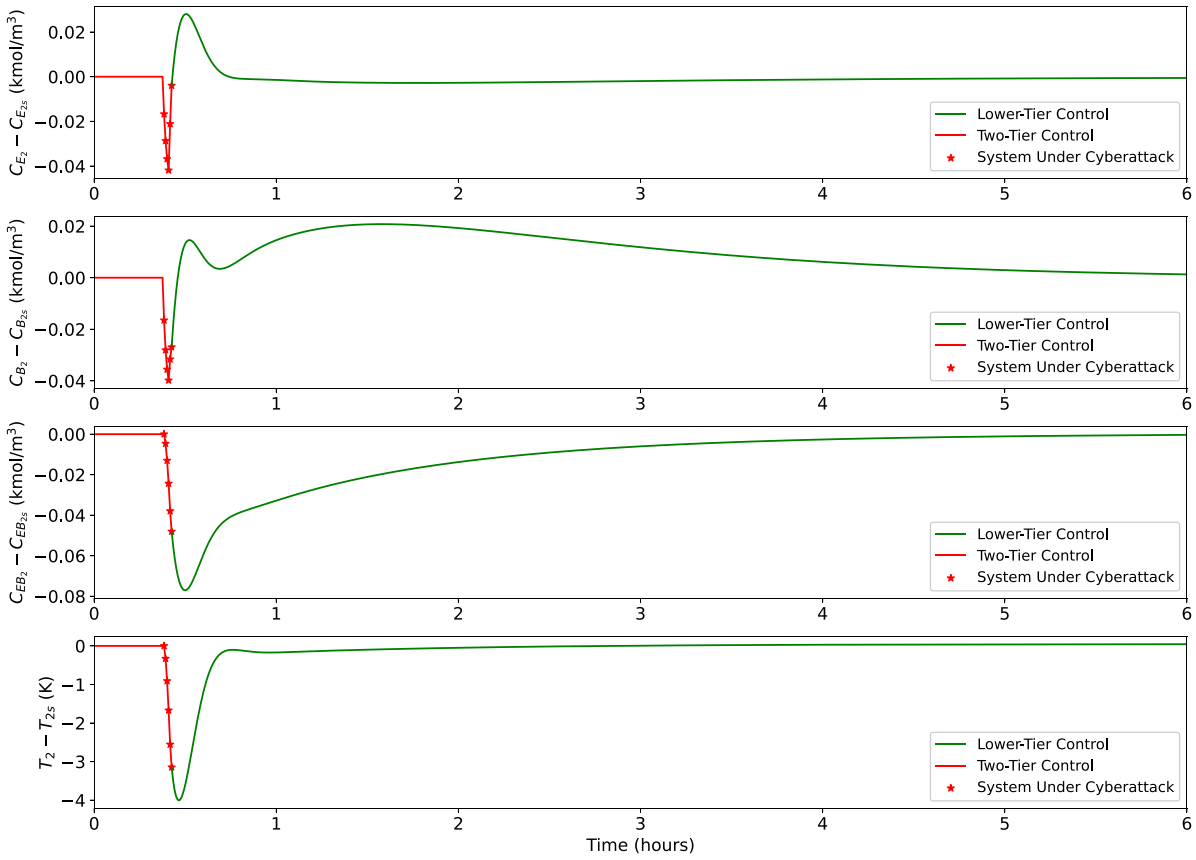

Fig. 9. State profiles of CSTR 2 under encrypted two-tier control (red line), encrypted two-tier control under cyberattack (red line with stars), and encrypted lower tier post-reconfiguration (green line), when a surge attack is launched on the upper-tier control inputs at $t = 23$ min.

encrypted control scheme without any cyberattack. At $t = 23$ min, a surge cyberattack is launched on all four control inputs of the upper tier by manipulating the MPC control inputs before they are encrypted. This manipulation deviates the system from its desired stability region, $\Omega_{\rho_{\min}}$, without complete destabilization. The cyberattack detector begins detecting the attack status at each sampling instance after 20 min, requiring data from the preceding 40 sampling instances (equivalent to 20 min of process time). The detector identifies the cyberattack for the first time at 25 min, 2 min after the attack was initiated on the upper tier control inputs. After three consecutive detections at 25, 25.5, and 26 min, the upper tier control is disabled at 26 min. Subsequently, only the secure, encrypted lower-tier control scheme is employed to guide the system back to its desired stability region, $\Omega_{\rho_{\min}}$.

For Figs. 11 to 13, during the initial 23 min of the process time, the state trajectories exhibit swift convergence towards their steady-states as they operate under the two-tier encrypted control scheme without any cyberattack. At $t = 23$ min, a geometric cyberattack is initiated, targeting all 6 concentration states of the upper tier. This attack involves manipulating the decrypted state values received by the MPC in their plaintext form, and it deviates the system states from their prior converging trajectory towards their steady-states. The cyberattack detection mechanism commences after 20 min of the process, necessitating data from the preceding 40 sampling instances (equivalent to 20 min). The cyberattack detector first identifies the cyberattack at the 26 min, 3 min after the cyberattack was initiated. After three consecutive detection instances at 26, 26.5, and 27 min, the upper-tier control scheme is disabled at 27 min. Subsequently, only the secure, encrypted lower-tier control scheme is employed to guide the system back to its desired stability region, $\Omega_{\rho_{\min}}$. Also, in this scenario, when the cyberattack is launched, the system is in the process of converging towards its steady state; it has not yet reached its final equilibrium. Importantly, the cyberattack detector remains active and can effectively identify attacks even during this transitional phase, as illustrated in Figs. 11 to 13.

Remark 8. The presence of quantization introduces some irregularities in the curves of certain control inputs and states. For example, in Fig. 10, noticeable bumps can be observed in the control input response corresponding to the rate of heat removal in CSTR 1. These bumps stem from the fact that the quantization error value is multiplied by the gains of the controller within an encrypted framework. As a result, these multiplicative effects generate bumps in the trajectories of control inputs. However, in the case of the rate of heat removal for CSTR 2, this phenomenon is not as apparent in the same figure. This is attributed to the significantly larger magnitude of the control input for CSTR 2. Similarly, this irregularity is not observed in the inlet concentration control inputs for the CSTRs, as these control inputs are quantized after the plain text computation by the MPC. This approach prevents the multiplication of quantized terms, thus mitigating the generation of bumps due to control input quantization. Although the quantization effects are less conspicuous in the case of inlet concentration control inputs, their discontinuous behavior resulting from quantized terms still exists. This effect is mitigated by selecting a higher quantization parameter. As a solution, a quantization parameter of $d = 8$ has been opted for all the simulations that are being presented. This choice of a higher quantization parameter helps alleviate the observed irregularities.

5.6. Computational time of ML-based detection compared to encryption–decryption

This subsection delves into the computational load implications of incorporating machine-learning-based cyberattack detection within the

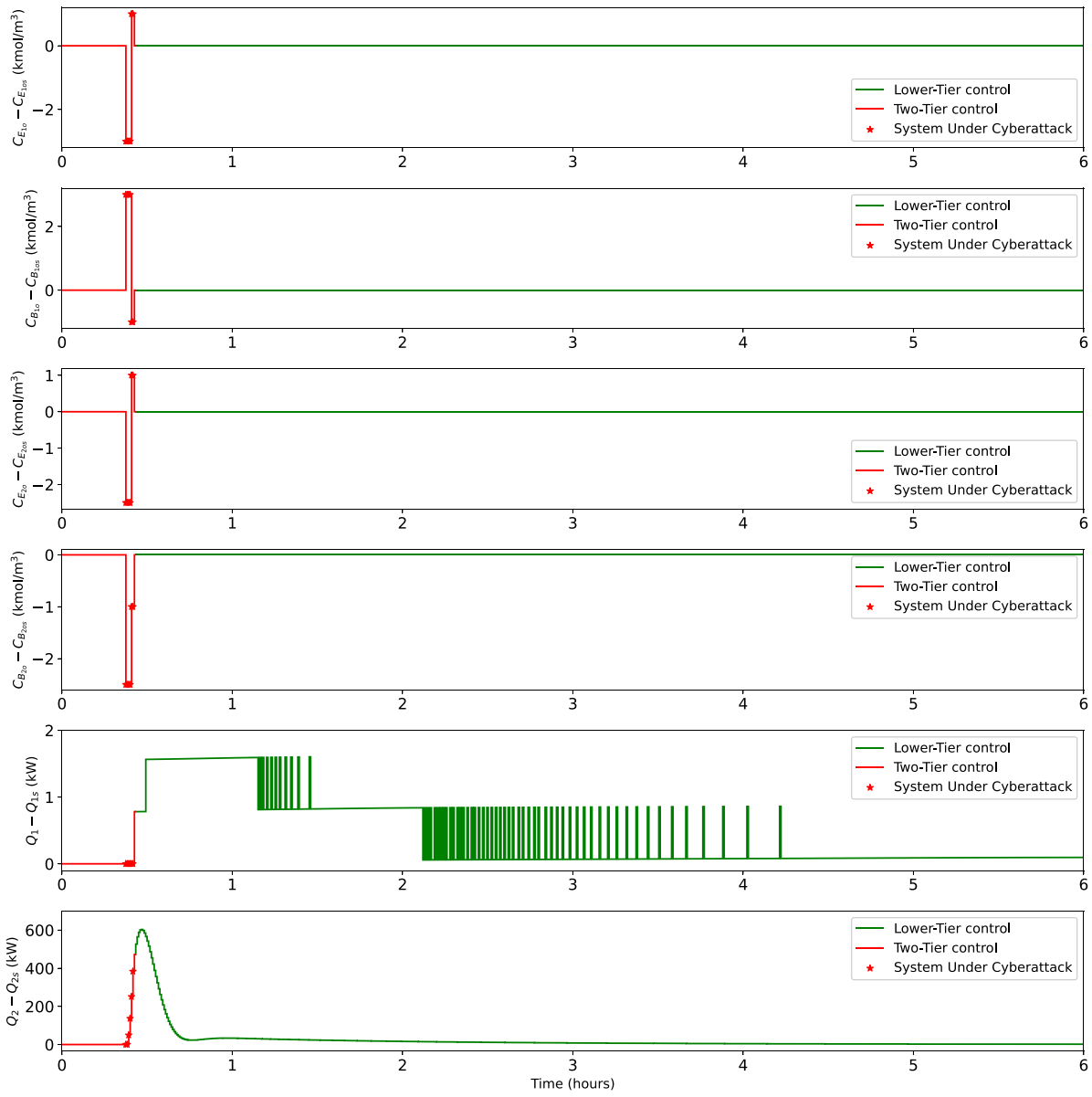

Fig. 10. Control input profiles under encrypted two-tier control (red line), encrypted two-tier control under cyberattack (red line with stars), and encrypted lower tier post-reconfiguration (green line), when a surge attack is launched on the upper-tier control inputs at $t = 23$ min.

encrypted control framework. A comparative analysis was conducted between the time dedicated to cyberattack detection and the time allocated to the encryption–decryption of upper-tier states and control inputs. The lower tier is fully secure as it maintains encrypted communication throughout the network (sensor–controller–actuator), thus rendering detection algorithms unnecessary for it. Due to the independent operation of the lower tier in relation to the upper tier, along with the redundancy of cyberattack detection for the lower tier, the time taken for encryption in it is excluded from this analysis.

Fig. 14 depicts the ratio of the time taken for cyberattack detection to the time required for encryption–decryption operations for 25 min of process time, corresponding to 50 consecutive sampling instances. It is evident from Fig. 14 that the ML-based cyberattack detection consumes, on average, less than 1% of the time required for encryption–decryption. Consequently, the integration of this detection mechanism does not impose a significant computational burden on the overall time complexity of the system. Instead, it introduces a crucial cybersecurity aspect, especially in situations where the encrypted upper tier might not be entirely cyber-secure due to the context in which plaintext data encryption or decryption occurs within the control architecture.

Remark 9. In this research, the lower tier of the two-tier encrypted control architecture functions as a secure, stabilizing feature in continuous operation throughout the process. When a cyberattack is detected, only the upper-tier is deactivated, while the lower tier continues to stabilize the system without any interruptions. Alternatively, in a different framework than the one proposed in this research, the lower tier controller can serve as a backup controller within the architecture if it is desired for the MPC to exclusively compute all control inputs. In such a scenario, when a cyberattack is detected, control would be transitioned from the upper tier to the lower tier. The lower tier remains inactive during normal process operation when no cyberattack is detected. Thus, at any time only one tier would be functional. Nevertheless, as previously stated, in this study, both the lower and upper tiers remain operational in the absence of any cyberattack. In the event of a detected cyberattack, the upper tier is deactivated, while the lower tier continues its role in stabilizing the system.

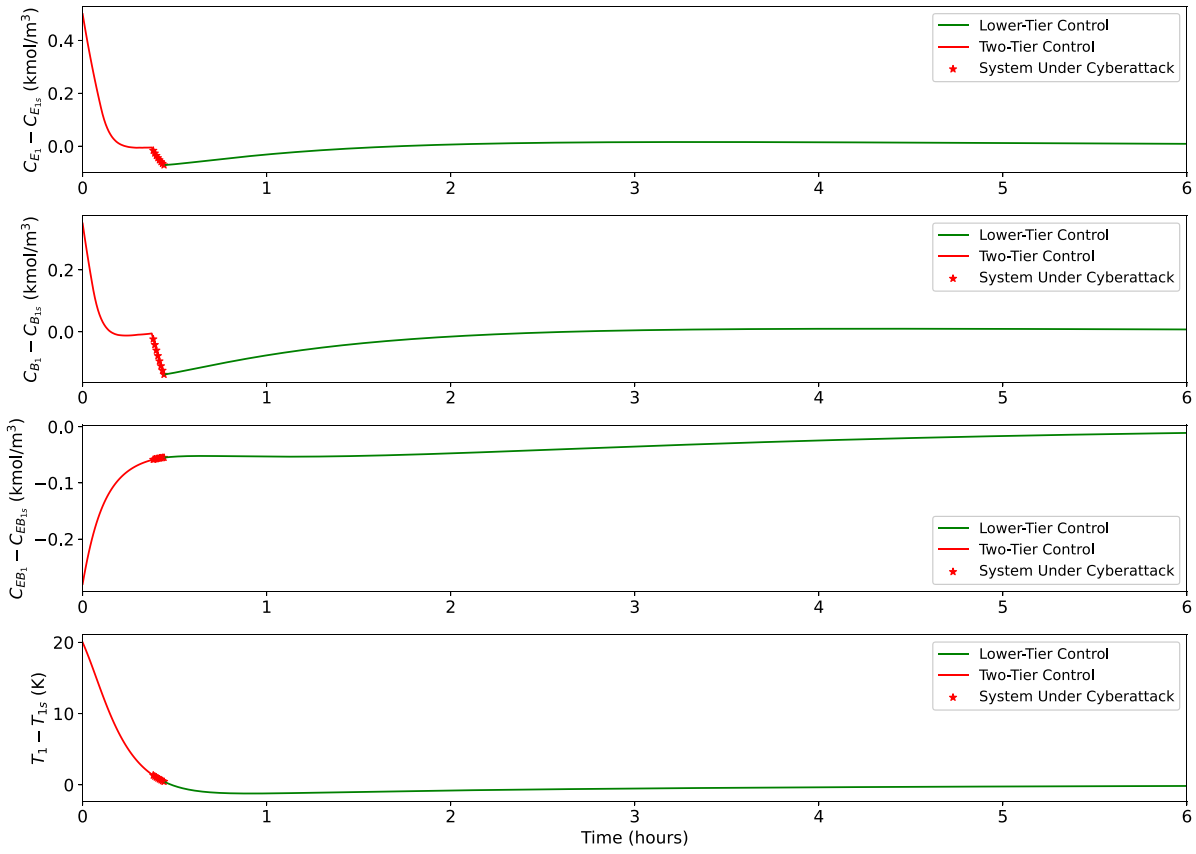

Fig. 11. State profiles of CSTR 1 under encrypted two-tier control (red line), encrypted two-tier control under cyberattack (red line with stars), and encrypted lower tier post-reconfiguration (green line), when a geometric attack is launched on the upper-tier states at $t = 23$ min.

6. Conclusions

In this study, we presented an encrypted two-tier control architecture incorporating an ML-based cyberattack detector to enhance the operational safety, cybersecurity, and closed-loop performance of nonlinear process systems. The lower-tier control system comprises a set of encrypted proportional–integral controllers, while the upper-tier control system employs an encrypted Lyapunov-based model predictive controller. This architecture enhances system cybersecurity, even in settings where control input computations may not be cybersecure. By integrating both linear and nonlinear controllers with encryption, the developed two-tier control architecture can be adapted to large-scale nonlinear processes. Further, we have provided insights into the framework and formulation of the encrypted lower- and upper-tier control systems. Through a comprehensive stability analysis, we have identified potential sources of error and established bounds to ensure closed-loop system stability. Additionally, we have delved into the development of an ML-based cyberattack detector, addressed critical aspects such as quantization parameter selection, sampling time criteria, and computational load assessment. These issues are essential for the practical implementation of the proposed control system across nonlinear processes. To validate the efficacy of our control framework, we subjected it to previously unseen cyberattack patterns within a nonlinear chemical process network utilized in ethylbenzene production. We carried out a detailed simulation study that exposed the implementation and performance of the two-tier control architecture and the usefulness of the cyberattack detector. In summary, our work advances control system cybersecurity by integrating ML-based cyberattack detection into encrypted control systems with both linear and nonlinear controllers.

CRediT authorship contribution statement

Yash A. Kadakia: Conceptualization, Methodology, Software, Manuscript writing. **Atharva Suryavanshi:** Conceptualization, Methodology, Software, Manuscript writing. **Aisha Alnajdi:** Conceptualization, Methodology, Software. **Fahim Abdullah:** Conceptualization, Methodology, Software. **Panagiotis D. Christofides:** Manuscript reviewing & editing.

Declaration of competing interest

The authors declare that they have no known competing financial interests or personal relationships that could have appeared to influence the work reported in this paper.

Data availability

Data will be made available on request.

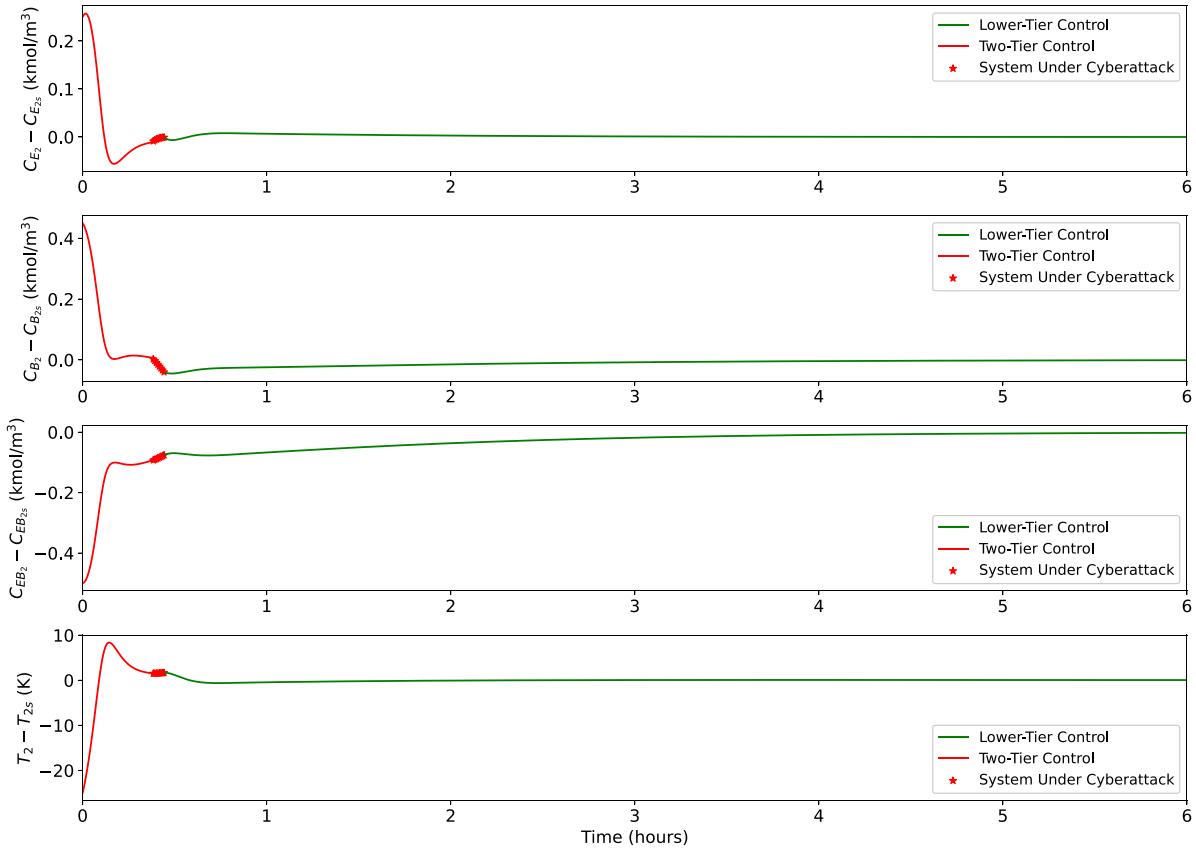

Fig. 12. State profiles of CSTR 2 under encrypted two-tier control (red line), encrypted two-tier control under cyberattack (red line with stars), and encrypted lower tier post-reconfiguration (green line), when a geometric attack is launched on the upper-tier states at $t = 23$ min.

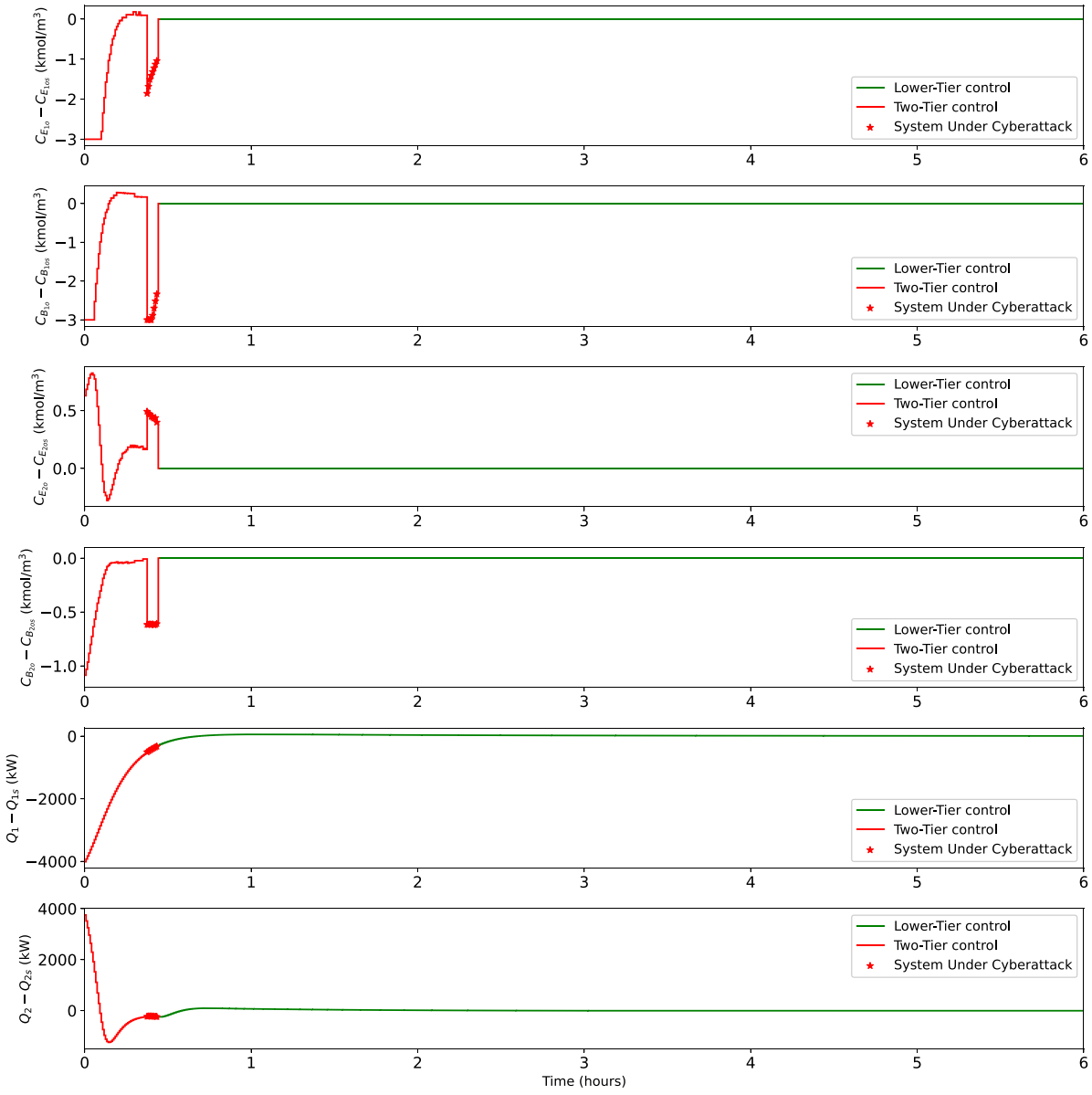

Fig. 13. Control input profiles under encrypted two-tier control (red line), encrypted two-tier control under cyberattack (red line with stars), and encrypted lower tier post-reconfiguration (green line), when a geometric attack is launched on the upper-tier states at $t = 23$ min.

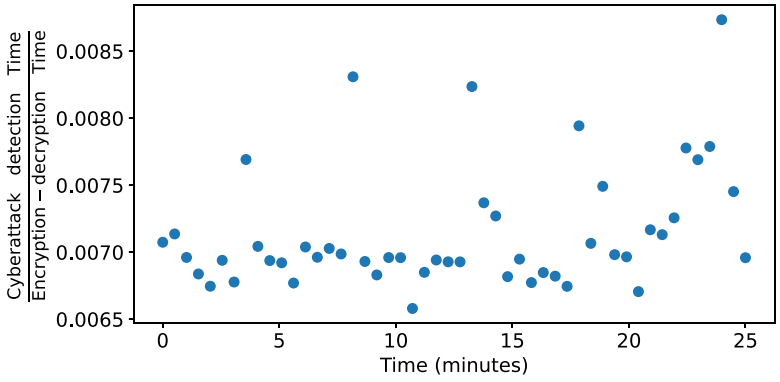

Fig. 14. Ratio of the time for ML-based Cyberattack detection to encryption–decryption for 50 consecutive sampling periods.

Acknowledgment

Financial support from the National Science Foundation, United States, CBET-2227241, is gratefully acknowledged.

References

Agrawal, S., Agrawal, J., 2015. Survey on anomaly detection using data mining techniques. Procedia Comput. Sci. 60, 708–713.

Bomze, I.M., Demyanov, V.F., Fletcher, R., Terlaky, T., 2010. Nonlinear optimization: lectures given at the CIME Summer School held in Cetraro, Italy, July 1-7, 2007. Springer.

Chen, S., Wu, Z., Christofides, P.D., 2020. A cyber-secure control-detector architecture for nonlinear processes. AIChE J. 66, e16907.

Darup, M.S., Redder, A., Shames, I., Farokhi, F., Quevedo, D., 2017. Towards encrypted MPC for linear constrained systems. IEEE Control Syst. Lett. 2, 195–200.

Data61, C., 2013. Python paillier library. https://github.com/data61/python-paillier.

Durand, H., 2018. A nonlinear systems framework for cyberattack prevention for chemical process control systems. Mathematics 6, 169.

Durand, H., Wegener, M., 2020. Mitigating safety concerns and profit/production losses for chemical process control systems under cyberattacks via design/control methods. Mathematics 8, 499.

Heidarinejad, M., Liu, J., Christofides, P.D., 2012. Economic model predictive control of nonlinear process systems using Lyapunov techniques. AIChE J. 58, 855–870.

Huang, L., Nguyen, X., Garofalakis, M., Hellerstein, J.M., Jordan, M.I., Joseph, A.D., Taft, N., 2007. Communication-efficient online detection of network-wide anomalies. In: Proceedings of 26th IEEE International Conference on Computer Communications. Barcelona, Spain, pp. 134–142.

Kadakia, Y.A., Suryavanshi, A., Alnajdi, A., Abdullah, F., Christofides, P.D., 2023. Encrypted model predictive control of a nonlinear chemical process network. Processes 11, 2501.

Khalil, H., 2002. Nonlinear Systems. In: Pearson Education, Prentice Hall.

Kushner, D., 2013. The real story of stuxnet. IEEE Spectr. 50, 48–53.

Liu, J., de la Peña, D.M., Ohran, B.J., Christofides, P.D., Davis, J.F., 2008. A two-tier architecture for networked process control. Chem. Eng. Sci. 63, 5394–5409.

Liu, J., Munoz de la Pena, D., Ohran, B.J., Christofides, P.D., Davis, J.F., 2010. A two-tier control architecture for nonlinear process systems with continuous/asynchronous feedback. Internat. J. Control 83, 257–272.

Mhaskar, P., El-Farra, N.H., Christofides, P.D., 2006. Stabilization of nonlinear systems with state and control constraints using Lyapunov-based predictive control. Systems Control Lett. 55, 650–659.

Narasimhan, S., El-Farra, N.H., Ellis, M.J., 2022a. Active multiplicative cyberattack detection utilizing controller switching for process systems. J. Process Control 116, 64–79.

Narasimhan, S., El-Farra, N.H., Ellis, M.J., 2022b. A control-switching approach for cyberattack detection in process systems with minimal false alarms. AIChE J. 68, e17875.

Narasimhan, S., El-Farra, N.H., Ellis, M.J., 2023. A reachable set-based scheme for the detection of false data injection cyberattacks on dynamic processes. Digit. Chem. Eng. 7, 100100.

Omar, S., Ngadi, A., Jebur, H.H., 2013. Machine learning techniques for anomaly detection: an overview. Int. J. Comput. Appl. 79, 33–41.

Paillier, P., 1999. Public-key cryptosystems based on composite degree residuosity classes. In: Proceedings of the International Conference on the Theory and Applications of Cryptographic Techniques. Springer, Berlin, Heidelberg, pp. 223–238.

Singh, G.G., Sajid, Z., Khan, F., Mather, C., Bernhardt, J., Frölicher, T., 2023. Rethinking disaster risk for ecological risk assessment. Front. Ecol. Evol. 11, 1249567.

Suryavanshi, A., Alnajdi, A., Alhajeri, M., Abdullah, F., Christofides, P.D., 2023. Encrypted model predictive control design for security to cyberattacks. AIChE J. 69, e18104.

Wächter, A., Biegler, L.T., 2006. On the implementation of an interior-point filter line-search algorithm for large-scale nonlinear programming. Math. Program. 106, 25–57.

Wu, Z., Albalawi, F., Zhang, J., Zhang, Z., Durand, H., Christofides, P.D., 2018. Detecting and handling cyber-attacks in model predictive control of chemical processes. Mathematics 6, 173.

Wu, Z., Chen, S., Rincon, D., Christofides, P.D., 2020. Post cyber-attack state reconstruction for nonlinear processes using machine learning. Chem. Eng. Res. Des. 159, 248–261.