Received: 26 September 2019 | Revised: 26 November 2019 | Accepted: 23 December 2019

DOI: 10.1002/aic.16907

PROCESS SYSTEMS ENGINEERING

AIChE JOURNAL

A cyber-secure control-detector architecture for nonlinear processes

Scarlett Chen[1] | Zhe Wu[1] | Panagiotis D. Christofides[1,2]

[1]Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, California

[2]Department of Electrical and Computer Engineering, University of California, Los Angeles, California

Correspondence
Panagiotis D. Christofides, Department of Chemical and Biomolecular Engineering, University of California, Los Angeles, CA 90095.
Email: pdc@seas.ucla.edu

Funding information
Department of Energy; National Science Foundation

Abstract

This work presents a detector-integrated two-tier control architecture capable of identifying the presence of various types of cyber-attacks, and ensuring closed-loop system stability upon detection of the cyber-attacks. Working with a general class of nonlinear systems, an upper-tier Lyapunov-based Model Predictive Controller (LMPC), using networked sensor measurements to improve closed-loop performance, is coupled with lower-tier cyber-secure explicit feedback controllers to drive a nonlinear multivariable process to its steady state. Although the networked sensor measurements may be vulnerable to cyber-attacks, the two-tier control architecture ensures that the process will stay immune to destabilizing malicious cyber-attacks. Data-based attack detectors are developed using sensor measurements via machine-learning methods, namely artificial neural networks (ANN), under nominal and noisy operating conditions, and applied online to a simulated reactor-reactor-separator process. Simulation results demonstrate the effectiveness of these detection algorithms in detecting and distinguishing between multiple classes of intelligent cyber-attacks. Upon successful detection of cyber-attacks, the two-tier control architecture allows convenient reconfiguration of the control system to stabilize the process to its operating steady state.

KEYWORDS

attack detection, cyber-attacks, model predictive control, neural networks, nonlinear processes, process control

1 | INTRODUCTION

Automated real-time operations of industrial process control systems depend heavily on accurate information and reliable communication technologies. These cyber-physical systems (CPS) utilize hardware and software resources to seamlessly integrate computation, network, and physical process components. In more recent years, wireless networks and internet communication are starting to complement or replace existing wired point-to-point communications, and together constitute a hybrid communication network.[1] As these new developments bring efficiency and improved closed-loop performance, heightened concern for security also arises.[2] As more components are included, there is a high probability that continuous feedback measurements cannot be guaranteed due to bursts of network transmission errors. Furthermore, each device and communication channel in the network expand the exploitable surface to cyber-attacks. With increasing sophistication of the cyber-attacks, the negative consequences associated with these attacks may be beyond asset damage and economic loss. Since attackers may have access to technical details of the control system and of the process operation, fundamental process safety and operational integrity may also be compromised. A number of industrial cyber-attacks in recent years, such as the Stuxnet worm attacking Iran's nuclear centrifuges, the 2014 cyber-attack targeting a German steel mill, and the 2015 cyber-attack on Ukraine's electric power grid, have all proven their detrimental physical impacts.[3] Current mitigation practice recommends

having multiple independent layers of protection, such as personnel training, network compartmentalization, access restriction, using "unhackable" analog backup, running anti-virus softwares, and so on.[2] However, there is no systematic approach to actively monitor, detect, and contain these intrusions using the data network on a digital platform. Therefore, designing new advanced detection schemes coupled with robust control architectures to cyber-attacks can provide a resilient solution to addressing this gap. Considering the close interaction between cyber and physical components, operational cyber-security of the control systems requires a different strategy from the traditional information technology (IT) approaches.

Recent IT developments have given an edge to enterprise cyber-security (e.g., enhancement of firewalls for guarding confidentiality), and similar safeguarding methodologies such as advanced big data analytics may also be used to secure device measurements, which are instrumental to process operation. Production plants collect and archive huge operational and instrumentation data to be used for monitoring, control, and troubleshooting. However, much of these big data archived is seldom revisited. The potential application of these data goes beyond preventative maintenance and fault detection, especially with increased digital connectivity and increased computing power. One example usage of big data is the detection and prediction of cyber-attacks in the plant. In recent years, machine-learning techniques have become increasingly popular in classical engineering fields in addition to computer science and engineering.[4-6] Artificial neural networks were trained to model a pilot-scale entrained-flow gasifier in Reference 7, and were also used to predict industrially relevant observable values and/or stochastic PDE model parameters for nonlinear model predictive control of multiscale thin film deposition processes in References 8, 9. Conventional machine-learning methods (e.g., artificial neural network, principal component analysis, support vector machines) and more advanced deep learning methods (e.g., convolutional neural networks, long short-term memory neural networks, gated recurrent units) have demonstrated success in detecting machine and plant anomalies.[10] Model-based fault diagnosis and classification in electric drive systems were carried out using a fault diagnostic neural network in Reference 11 and automated fault detection and diagnosis of HVAC subsystems using hidden Markov models was studied in Reference 12. Moreover, machine-learning methods deployed for attack detection were presented in a number of works.[13-15] Particularly, the anomaly detection algorithm outlined in Reference 16 used a long short-term memory (LSTM) neural network as a predictor to model normal behavior of a water treatment testbed, and used the Cumulative Sum (CUSUM) method to identify anomalies. Using various machine-learning classification methods, cyber-attacks on power systems were distinguished from process disturbances in Reference 17, and a behavior-based intrusion detection algorithm was developed to identify the type of attack.[18] Similarly, detection of cyber-attacks in a chemical process was realized via development of feed-forward artificial neural networks in Reference 19, where compromised signals were rerouted to a secure sensor upon detection. These recent literature contributions have demonstrated the feasibility of machine-learning algorithms in anomaly detection. Therefore, machine-learning methodologies can be readily adopted in the context of control theory and cyber-physical security. Stealthy, intelligent cyber-attack diagnosis and defense span a much broader scope than classical fault detection problems because intelligent adversaries can modify the actuator, the sensor, or the control implementation using process and control system information.

With knowledge of the plant model and of the control formulation, cyber-attacks are strategically programmed with the goal of disruption, and are fundamentally different from ordinary sensor and actuator faults. Specifically, among sensor cyber-attacks, Denial-of-Service attacks, replay attacks and deception attacks (e.g., Min-Max, Geometric, Surge) are some of the most common and easily implementable ones by attackers.[20] Furthermore, the effects of these attacks may be only observed in changes of the dynamic behavior (runtime variables) of the closed-loop system; thus, using hardware performance counters to track code modifications is not feasible.[3] While conventional detection methods have demonstrated their effectiveness in detecting suspicious process variable deviations, most of these methods are model-based—either dependent on network and computer system models, or on physical process models. Certain classes of intelligent cyber-attacks either render traditional detection methods ineffective, or remain undetected until the system experiences a significant deviation and reaches an undesirable operating point, at which the existing alarm systems could be triggered. The goal of a robust cyber-attack detector is to identify attacks from subtle variations in real-time process state measurements and mitigate the risk before an operation alarm is triggered. Therefore, without explicit knowledge on the process model, adopting a databased detection approach utilizing machine-learning algorithms provides a promising path for the detection of unknown intelligent cyber-attacks. The integration of existing advanced control techniques (e.g., MPC) and online machine-learning-based detection algorithms adds another protective safeguard to the multilayer cyber-defense strategy that is standard to next-generation smart manufacturing.

Despite current literature efforts on stealthy attack analysis and machine-learning-based detection, there is a lack of an integration of the two, as well as a broader application of detection schemes across stealthy attack classes and nonlinear chemical processes. Furthermore, feasible mitigation practices using control strategies after the occurrence of attacks have not yet been explored. In light of these gaps, the contributions of this work are as follows: (a) construction of data-based machine-learning detection algorithms which can effectively detect multiple classes of intelligent cyber-attacks; (b) design of a robust control architecture to promptly contain and eliminate the impact of cyber-attacks by reconfiguring the control system; and (c) application of the proposed detection and mitigation schemes to a benchmark multivariable nonlinear process example, which is a process example widely used in literature to test the performance of new control system designs.[21-23] The remainder of this paper is organized as follows: notation and the class of nonlinear process systems considered are presented in Section 2; the cyber-secure control architecture is formulated in Section 3; the design and detection mechanism of cyber-attacks are presented in Section 4; and the application of the

proposed methodology to a nonlinear chemical process network is presented in Section 5.

2 | PRELIMINARIES

2.1 | Nonlinear system formulation

In this work, $|\cdot|$ is used to denote the Euclidean norm of a vector; x^T denotes the transpose of x; $\mathbf{R}^n_+$ denotes the set of vector functions of dimension n whose domain is $[0, \infty)$. Class $\mathcal{K}$ functions $\alpha(\cdot): [0, a) \rightarrow [0, \infty]$ are defined as strictly increasing scalar functions with $\alpha(0) = 0$. The class of continuous-time nonlinear systems considered is described by the following state-space form:

$$\dot{x}(t) = f(x(t), u_c(t), u_a(t)) \tag{1a}$$

$$y_c(t) = h_c(x(t)), y_a(t) = h_a(x(t)) \tag{1b}$$

where $x \in \mathbf{R}^{n_x}$ is the state vector, $y_c(t) \in \mathbf{R}^{n_{yc}}$ represents the vector of state measurements that are sampled continuously (e.g., reactor temperature), and $y_a(t) \in \mathbf{R}^{n_{ya}}$ represents the vector of networked state measurements that may be sampled asynchronously at $t = t_k$ (e.g., reactor product concentration); u_c and u_a are the manipulated input vectors, which are constrained by $[u_c \in \mathbf{R}^{m_{u_c}}, u_a \in \mathbf{R}^{m_{u_a}}] \in U := \{u_i^{\min} \le u_i \le u_i^{\max}, i = 1, \ldots, m_{u_c} + m_{u_a}\}$. Through y_c and y_a, we assume measurement of the full state vector x can be obtained at t_k. Without loss of generality, the initial time t_0 is taken to be zero ($t_0 = 0$). It is assumed that $f(\cdot)$ is a sufficiently smooth vector function of its arguments, and $h_c(\cdot)$ and $h_a(\cdot)$ are sufficiently smooth vector functions of x where $f(0, 0, 0) = 0$, $h_c(0) = 0$, $h_a(0) = 0$. Thus, the origin is an equilibrium point of the system of Equation (1) under $u_c(t) = 0$ and $u_a(t) = 0$.

3 | CYBER-SECURE TWO-TIER CONTROL ARCHITECTURE

We propose a cyber-secure control architecture that unites a lower-tier control system that uses the dedicated sensor measurements, $y_c(t)$, to ensure stability of the steady state of the closed-loop system and an upper-tier, high-performance control system (in this work, model predictive control) that uses both dedicated ($y_c(t)$) and networked ($y_a(t)$) sensor measurements to improve closed-loop performance significantly above what could be achieved with the lower-tier control system. Below we present in detail the lower-tier and upper-tier control systems.

3.1 | Lower-tier control system

We assume that there exists an explicit feedback controller of the form $u_c(t) = \phi_c(y_c(x(t))) \in U$ that can stabilize the closed-loop system of Equation (1). This controller, using only the continuous measurements $y_c(t)$ is termed the lower-tier controller, and is designed such that the origin of the nominal closed-loop system of Equation (1) with the input $u_a(t) = 0$ is rendered asymptotically stable. Therefore, there exist class $\mathcal{K}$ functions $\alpha_i(\cdot)$, $i = 1, 2, 3, 4$, and a positive definite control Lyapunov function $V(x)$ that satisfy the following conditions:

$$\alpha_1(|x|) \le V(x) \le \alpha_2(|x|), \tag{2a}$$

$$\frac{\partial V(x)}{\partial x} f(x, \phi_c(y_c(x)), 0) \le -\alpha_3(|x|), \tag{2b}$$

$$\left|\frac{\partial V(x)}{\partial x}\right| \le \alpha_4(|x|) \tag{2c}$$

for all $x \in D \subseteq \mathbf{R}^{n_x}$, where D is an open neighbourhood around the origin. We construct a subset defined as a level set of $V(x)$ inside D, $\Omega_\rho := \{x \in D \mid V(x) \le \rho, \rho > 0\}$, to represent an estimate of the stability region of the closed-loop system of Equation (1) under $\phi_c(y_c)$. Ω_ρ is an invariant set for the closed-loop system. Therefore, starting from any initial state in Ω_ρ, $\phi_c(y_c)$ guarantees that the state trajectory of the closed-loop system remains within Ω_ρ and asymptotically converges to the origin. Thus, given that the sensor measurements received by the lower-tier controller are secure and reliable, the lower-tier controller is able to stabilize the process to the origin for any initial conditions inside Ω_ρ.

3.2 | Upper-tier model predictive control system

To fully utilize the networked (potentially asynchronous) state measurements $y_a(t)$ and to compute $u_a(t)$ that improves the overall closed-loop performance over what can be achieved with $\phi_c(y_c)$, a Lyapunov-based MPC (LMPC) is used as the upper-tier controller with its contractive constraint defined based on the stability region of the lower-tier controller such that the asymptotic stability of the closed-loop system will not be jeopardized by the contributions of $u_a(t)$. The optimization problem of LMPC is as follows:

$$\mathcal{J} = \min_{u_a \in S(\Delta)} \int_{t_k}^{t_{k+N}} L(\tilde{x}(t), \tilde{u}_c(t), u_a(t)) dt \tag{3a}$$

$$\text{s.t. } \dot{\tilde{x}}(t) = f(\tilde{x}(t), \phi_c(y_c(\tilde{x}(t)), u_a(t)) \tag{3b}$$

$$\dot{\hat{x}}(t) = f(\hat{x}(t), \phi_c(y_c(\hat{x}(t)), 0) \tag{3c}$$

$$\tilde{x}(t_k) = \hat{x}(t_k) = x(t_k) \tag{3d}$$

$$[u_c(t) u_a(t)] \in U, \forall t \in (t_k t_{k+N}) \tag{3e}$$

$$V(\tilde{x}(t_k)) \le V(\hat{x}(t_k)), \text{ if } V(\tilde{x}(t_k)) > \rho_{\min} \tag{3f}$$

$$V(\tilde{x}(t)) \le \rho_{\min}, \forall t \in (t_k t_{k+N}), \text{ if } V(\tilde{x}(t_k)) \le \rho_{\min} \tag{3g}$$

where u_a belongs to a family of piece-wise constant functions $S(\Delta)$ with sampling period Δ, N is the number of sampling periods in the

prediction horizon, and the LMPC optimization problem presented in Equation (3) optimizes u_a over the prediction horizon $t \in [t_k, t_{k+N})$ when full state measurement is received at time instance t_k. The optimal solution is denoted $u_a^*(t)$. The first control action of $u_a^*(t)$, i.e., $u_a(t) = u_a^*(t_k)$, is applied in open loop until a new full-state measurement $x(t_k)$ obtained from $y_c(t_k)$ and $y_a(t_k)$ in Equation (1b) becomes available to the LMPC and the optimization problem is solved again. In the meantime, the lower-tier controller continuously calculates and applies $u_c(t) = \phi_c(y_c(t))$ based on continuous measurement feedback $y_c(t)$. If the time between two consecutive asynchronous measurements is longer than the prediction horizon $N \cdot \Delta$, then $u_a(t)$ is set to zero for the remainder of the asynchronous sampling interval past the prediction horizon, such that it does not act as an additional disturbance to the lower-tier controller before the next y_a arrives. In Equations (3b) and (3c), $\tilde{x}(t)$ and $\hat{x}(t)$ are the predicted state trajectories of the two-tier nominal system using control actions $\phi_c(y_c(\tilde{x}(t)))$ coupled with $u_a(t)$ computed by LMPC, and control actions $\phi_c(y_c(\hat{x}(t)))$ coupled with $u_a(t) = 0$, respectively. As shown in Equation (3d), full-state measurements received at t_k are used as the initial conditions of the predicted trajectories in the optimization problem of LMPC. Both upper-tier and lower-tier controller inputs are subject to their respective constraints defined by U in Equation (3e).

Given that the lower-tier controller is able to stabilize the system independently as the Lyapunov function under lower-tier control satisfies the conditions in Equation (2), the contractive constraint in Equation (3f) ensures that the value of the Lyapunov function of the closed-loop system under two-tier control, $V(\tilde{x}(t_k))$, is lower than or equal to that under lower-tier control alone $V(\hat{x}(t_k))$. Therefore, Ω_ρ is the closed-loop stability region under two-tier control. In other words, the upper-tier maintains the closed-loop stability of the system while improving the overall closed-loop performance. In order to avoid oscillations when the states approach the equilibrium point, the Lyapunov function is bounded as seen in Equation (3g) once the system enters a small region around the equilibrium point characterized by a level set $\Omega_{\rho_{\min}}$, where $0 < \rho_{\min} < \rho$. It is important to note that the LMPC is only executed when full-state information is received from both the continuous and asynchronous measurements as they become available at time t_k. The continuous measurements are measured and transmitted by sensors in a point-to-point network, and are used by the lower-tier control system to compute stabilizing control actions continuously. Thus, the continuous measurements will be available and readily used as state feedback in addition to the asynchronous measurements when the LMPC is activated. This two-tier control design, where the networked sensor measurements, $y_a(t)$, used only by the upper-tier controller may be under potential cyber-attack, is illustrated in Figure 1a.

4 | CYBER-ATTACK DESIGN AND DETECTION

4.1 | Attack scenarios

The upper-tier control system—where networked sensor information is incorporated and part of its measurement feedback is asynchronous—is vulnerable to cyber-attacks. Due to these irregular and sparse measurements, suspicious disparities between consecutive state measurements may be less apparent or susceptible to detection by the control engineer or classical fault-detection schemes. Furthermore, deviations caused by intelligent cyber-attacks and their dynamic impact on the process may be less detectable when multiple states are attacked. While the networked asynchronous measurements used only in the upper-tier controller are more susceptible to attacks, the continuous measurements used in both upper-tier and lower-tier controllers must remain intact for a few reasons. First, we assume that the process is stabilized under lower-tier controllers, and the upper-tier controller is designed such that its control Lyapunov function is contained inside that of the stabilizing lower-tier controllers. Therefore, the closed-loop stability under two-tier control is ensured by the stabilizing lower-tier controllers, and the closed-loop stability under lower-tier control is only guaranteed if the continuous measurements feeding into lower-tier controllers are secure and reliable. Second, having a secure stabilizing lower-tier controller allows quick mitigation measures by changing the control structure once a cyber-attack is identified in the networked measurements. Since the process can be driven to its operating steady state using only the lower-tier control system, in the case of a confirmed cyber-attack detection, the upper-tier controller which uses the corrupted networked measurements will be shut off. If the continuous measurements are also tampered, then the closed-loop stability under the lower-tier controllers is no longer guaranteed, and this mitigation plan is rendered ineffective. Therefore, having secure continuous sensor measurements is instrumental to maintaining functional stabilizing lower-tier controllers, which in turn ensures robustness of the closed-loop system to cyber-attacks.

To capture realistic sensor variance and to differentiate cyber-attacks from normal device fluctuations, bounded sensor noise is also considered. Thus, in our formulation, two scenarios are considered:

1. Nominal model is as presented in the nonlinear system outlined in Equation (1) where output sensors do not encounter any sensor noise.
2. Noise model adopts the same dynamic system model in Equation (1a), but with bounded Gaussian noise $w(t) \in W$ added to all sensor measurements, where $W = \left\{ w \in \mathbf{R}^{n_{y_c}+n_{y_a}} \mid |w| \le w_{\max} | \right\}$. Depending on the range of the outputs, the standard deviation of noise distribution for each sensor is adjusted accordingly. Therefore, Equation (1b) is modified to the following form:

$$y_c(t) = h_c(x(t)) + w(t),\ y_a(t) = h_a(x(t)) + w(t),\ w(t) \in W \tag{4}$$

To collect closed-loop data used for machine-learning detector training, attacks with varying durations L_a are introduced at random times i_0 during the simulation period. In both cases (with and without noise), signals without attack interceptions are classified as "no attack". Attacks on single sensor and on multiple sensors are both considered, where the data collected from single-sensor tampering is used for detector training, the multiple-sensor attacks are simulated

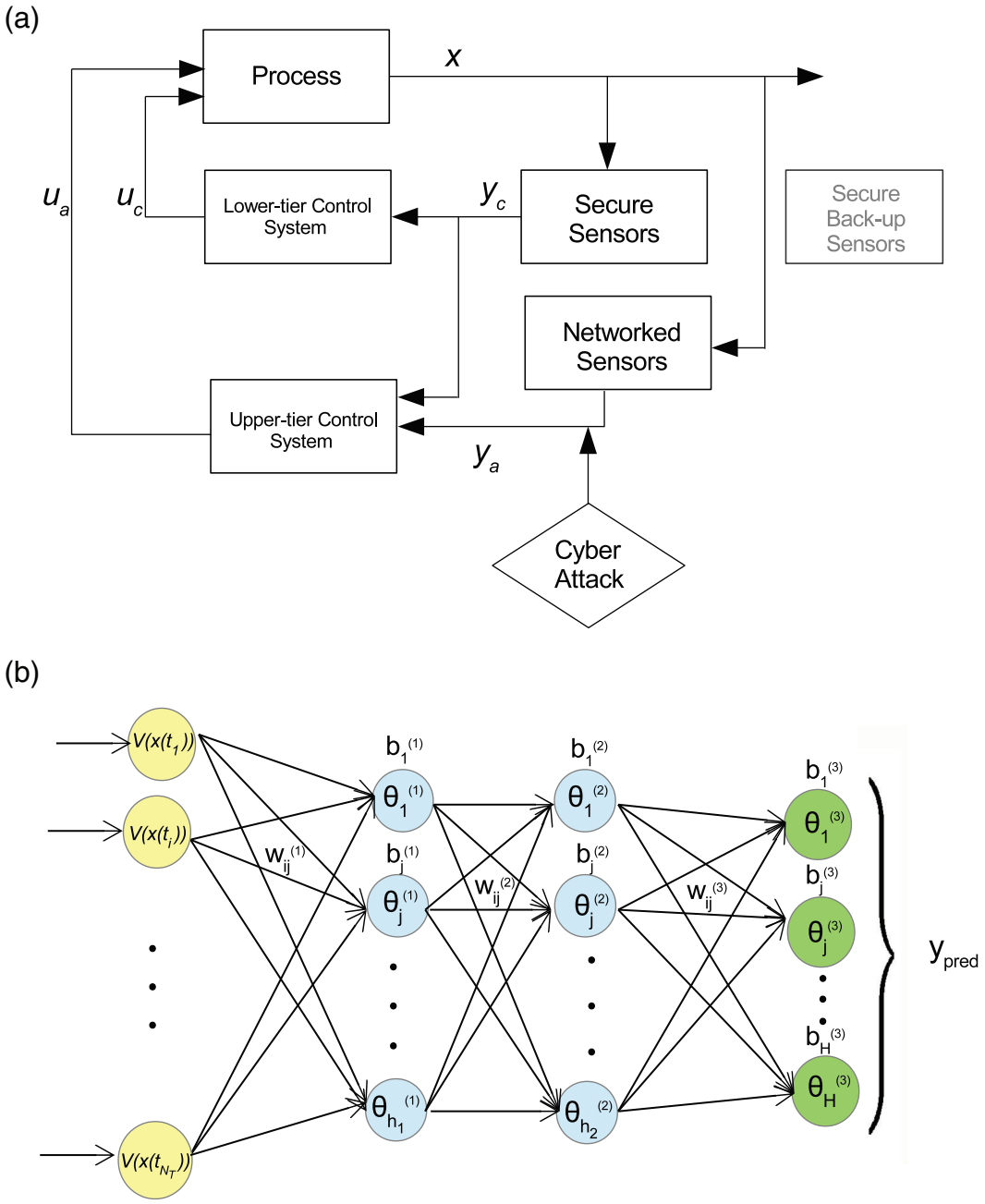

FIGURE 1 Two-tier control-detector architecture showing (a) lower-tier controllers using continuous secure sensor measurements and an upper-tier model predictive controller using both continuous (secure) and networked (vulnerable to cyber-attacks) sensor measurements, and (b) feed-forward neural network structure with two hidden layers with inputs being the full-state Lyapunov function at each sampling time of the model predictive controller within the detection window, and output being the probability of each class label for the examined trajectory indicating the status and/or type of cyber-attack [Color figure can be viewed at wileyonlinelibrary.com]

for online testing of the effectiveness of the detector algorithms. Data collected from single-sensor tampering also allows for sensor isolation using machine-learning-based models. For clarity, we consider that only one type of cyber-attack will occur at a time, that is, there will not be a hybrid of multiple attack types within each attack duration.

Remark 1 *The continuous state measurements used by lower-tier controllers are also used by the upper-tier controller despite the asynchronous execution frequency of the upper-tier controller. Since these continuous state measurements cannot be tampered, the same measurements fed into the upper-tier controller remain intact when an attack occurs. Thus, even in the case where multiple sensors are under attack, only those sending sampled asynchronous state measurements to the upper-tier controller will be corrupted. Moreover, it is not meaningful to simulate an intelligent cyber-attack that targets the two separate communication*

channels going into the two tiers of controllers such that the continuous measurements received by lower-tier controllers are accurate while the continuous measurements received by the upper-tier controller are falsified. This is because a simple tracker that examines the deviation between the same measurements received by both controllers would identify the presence of this abnormality. Therefore, the continuous state measurements remain unattacked in both controllers.

4.2 | Types of intelligent cyber-attacks

As intelligent cyber-attacks are adaptive to the process and control system behavior, we may assume that they are as powerful as having access to the measurement feedback signals (sensor attack), the control command signals (actuator attack), or auxiliary information such as the threshold and bias parameters in detection methods such as CUSUM.[24,25] Being process and controller behavior aware, the attacks will therefore have information on the stability region of the process under two-tier control, as well as existing alarm triggers on the ideal operating window imposed on the input and output variables. In this work, we only consider attacks on sensor measurements. During normal operation, these sensor feedback measurements need to accurately reflect the true state of the plant, otherwise any falsified measurement may result in control actions that no longer guarantee closed-loop stability, and may eventually drive the process away from its steady state and outside of Ω_ρ. Intelligent cyber-attacks are designed such that the controller is able to compute feasible control actions (i.e., the falsified state is not outside the closed-loop stability region Ω_ρ), but have large enough magnitude of variations such that the control system will not be able to drive the process to its operating steady state. The four most important types[25] of such attacks are considered below.

4.2.1 | Min-max cyber-attack

Min-max attacks are designed to induce maximum destabilizing impact within shortest time without being detected. In order to stay undetectable by classical detection methods, min-max attacks are introduced based on the more conservative value of the following two conditions: (a) a window around the equilibrium point of the attacked state(s) reflecting reasonable physical operating conditions; (b) state values furthest from the equilibrium point (minimum or maximum) such that the system does not exit the closed-loop stability region Ω_ρ. Attacks generated based on these two conditions ensure that the attacked state measurements fed to the control system do not exit the stability region or the configured operating window, and do not trigger any conventional detection alarms designed based on these boundary values. The min-max attack can be formulated as follows:

$$\bar{x}(t_i) = \min\left\{ \underset{x\in\mathbf{R}^{n_x}}{\operatorname{argmax}}\{V(x(t_i)) \le \rho\}, \underset{x\in\mathbf{R}^{n_x}}{\operatorname{argmax}}\{x(t_i)\in\chi\} \right\},\ \forall i\in[i_0, i_0+L_a] \tag{5}$$

where ρ defines the level set of the Lyapunov function $V(x)$ that characterizes the stability region of the closed-loop system of Equation (1) under the two-tier control system, $\chi := \{x_l \le x \le x_u\}$ represents the ideal state operating window, $\bar{x}$ is the compromised sensor measurement at each sampling step, i_0 marks the time instant that attack is added, and L_a denotes the time duration of the attack in terms of sampling periods.

4.2.2 | Replay cyber-attack

In a replay attack, the attacker first records segments of the system output corresponding to a nominal operating condition where large oscillations occur. The attacker then intercepts and resets the current process state measurements to these pre-recorded values. Replay attacks can be represented by the following equations:

$$\bar{x}(t_i) = x(t_k), \forall k\in[k_0, k_0+L_a],\ \forall i\in[i_0, i_0+L_a] \tag{6}$$

where $x(t_k)$ is the true plant measurement, L_a represents the length of the attack in terms of sampling periods, and $\bar{x}$ is the series of replay attacks introduced at time t_{i_0} duplicating previous plant measurements that are recorded starting from time t_{k_0}. As previous plant outputs are obtained from legitimate closed-loop measurements and given by secure sensors, these state values are supposedly inside the stability region and the operating envelope. Therefore, by replicating these values and feeding them back to the controller, classical detectors will not be able to recognize the abnormality caused by replay cyber-attacks.

4.2.3 | Geometric cyber-attack

Geometric cyber-attacks aim to deteriorate the closed-loop system stability slowly at the beginning, then geometrically increase their impact as time progresses, with its maximum damage achieved at the end of the attack duration. Initially, the attacker adds a small constant β to the true measured output (β is well below the maximum allowable value as defined in a min-max attack). At each subsequent time step, this offset is multiplied by $(1+\alpha)$, where $\alpha \in (0, 1)$, until it reaches the maximum allowable attack value. The two parameters α and β are therefore selected based on the stability region, the operating envelope, and the attack duration. Geometric attacks can be written in the form:

$$\bar{x}(t_i) = x(t_i) + \beta \times (1+\alpha)^{i-i_0},\ \forall i\in[i_0, i_0+L_a] \tag{7}$$

where $\bar{x}$ is the compromised sensor measurement, β and α are parameters that define the magnitude and speed of the geometric attack, i_0

signifies the time instant at which the attack starts, and L_a is the duration of the attack in terms of sampling periods.

4.2.4 | Surge cyber-attack

Surge attacks act similarly as min-max attacks initially to maximize the disruptive impact for a short period of time, then they are reduced to a lower value. In our case, the duration of the initial surge in terms of sampling times is selected as $L_s \in [2, 5]$ to differentiate itself from a min-max attack. Moreover, the initial surge period L_s is chosen to be in this range such that the potential time delays on the detection alarm will not be longer than L_s where the impact is most severe, and in turn, may cause a missed alarm from the detector. After the initial surge, the reduced constant value—at which the attack stays at—is chosen considering the impact of the initial surge and the total duration of the attack such that the cumulative error between state measurements and their steady state values will not exceed the threshold defined in some statistic-based detection methods (e.g., CUSUM).The formulation of a surge attack is presented below:

$$\begin{aligned}\bar{x}(t_i) &= \min\left\{\underset{x\in\mathbf{R}^{n_x}}{\operatorname{argmax}}\{V(x(t_i)) \le \rho\} \underset{x\in\mathbf{R}^{n_x}}{\operatorname{argmax}}\{x(t_i)\in\chi\}\right\}, \text{ if } i_0 \le i \le i_0 + L_s \\ \bar{x}(t_i) &= \underset{x\in\mathbf{R}^{n_x}}{\operatorname{argmax}}\{|x(t_i)|, 0 \le i \le i_0\}, \text{ if } i_0 + L_s < i \le i_0 + L_a\end{aligned} \tag{8}$$

where i_0 is the start time of the attack, L_s is the duration of the initial surge, and L_a is the total duration of the attack in terms of sampling periods. After the initial surge, the attack is reduced to a lower constant value, which is obtained by examining the secure state measurements prior to the occurrence of the surge attack, and taking the value that is furthest away from the origin.

4.3 | Machine-learning-based detection of cyber-attacks

There are many advantages to using a databased approach to develop the cyber-attack detector.[26-28] First, with attacks having possible access to information on process behavior (stability region and variable operating window), physical-model-based detection methods where the statistics threshold and false alarm bias are selected based on process operation are rendered ineffective.[25] Second, during real-life operation, plant model structure and parameters may be subject to modifications due to a changing operating environment. Therefore, using a data-based (physical model independent) method to train a detection mechanism for cyber-attacks is resilient to both process changes and intelligently designed attacks.

Within well-practiced machine-learning methods, neural networks (NN) have proven their effectiveness in both supervised and unsupervised classification problems.[29] Depending on the training data and target number of classes the algorithm aims to identify, neural networks can be used to distinguish between "attack" and "no attack" (two classes), or to identify the type of attack (multiple classes). While under attack, data collected from individual sensors can also be used to locate the corruption, where the neural network model distinguishes between multiple classes with each class representing one problematic sensor. In our study, a feed-forward artificial neural network is used for supervised classification. Through a series of nonlinear transformations, neurons in the first hidden layer are derived from the inputs, and hidden neurons in subsequent layers are derived from their precedent layer, with the output calculated from neurons in the last hidden layer. These nonlinear transformations are in the form of an activation function of biases and weighted sum of inputs (or neurons in the previous layer). The structure of a basic neural network model employed here is shown in Figure 1b, with each input representing the control Lyapunov function of the full state measurements at each asynchronous sampling time instant, and an output vector for predicted class label. The mathematical formulation of the two-hidden-layer feed-forward neural network is as follows:

$$\theta_j^{(1)} = g_1\left(\sum_{i=1}^{N_T} w_{ij}^{(1)} V(x(t_i)) + b_j^{(1)}\right) \tag{9a}$$

$$\theta_j^{(2)} = g_2\left(\sum_{i=1}^{h_1} w_{ij}^{(2)} \theta_i^{(1)} + b_j^{(2)}\right) \tag{9b}$$

$$\theta_j^{(3)} = g_3\left(\sum_{i=1}^{h_2} w_{ij}^{(3)} \theta_i^{(2)} + b_j^{(3)}\right),\ y_{pred} = \left[\theta_1^{(3)}, \theta_2^{(3)}, \ldots, \theta_H^{(3)}\right]^T \tag{9c}$$

with $\theta_j^{(1)}$ and $\theta_j^{(2)}$ representing neurons in the first and second hidden layer, respectively, where $j = 1, \ldots, h_l$ is the number of neurons in layer $l = 1$ and $l = 2$. $\theta_j^{(3)}$ represents neurons in the output layer ($l = 3$), where $j = 1, \ldots, H$, and H is the number of class labels. In this study, the number of hidden layers is 2; however, the formulation of neurons can be similarly applied to multiple hidden layers as well. In the input layer, input variables $V(x[t_i])$ are the control Lyapunov function of the full state measurements at time t_i, where $i = 1, \ldots, N_T$ is the length of the time-varying trajectory for each input sample. The weight associated with the connections between neurons i and j in consecutive layers (from $l-1$ to l) is denoted by $w_{ij}^{(l)}$, and the bias placed on the jth neuron in the lth layer is denoted by $b_j^{(l)}$. Each layer receives information from its previous layer, and computes an output based on the optimized weights, biases, and its nonlinear activation function—denoted g_l (e.g., hyperbolic tangent sigmoid transfer function $g(z) = \frac{2}{1+e^{-2z}} - 1$, and softmax function $g(z_j) = \frac{e^{z_j}}{\sum_{i=1}^{H} e^{z_j}}$ where H is the number of class labels). Performances of different common activation functions including ReLu, sigmoid, radial basis functions were analyzed in Reference 30. In the output layer, y_{pred} is a vector providing the predicted probabilities of each class label for the examined sample, where the neuron with the highest probability indicates the predicted class label. Depending on the type of classification problem the neural network is intended for, the predicted class label provides information on either the status or the type of a cyber-attack. The weights and biases are optimized by minimizing the Bayesian

regularized mean squared error cost function. The cost function used in the optimization problem is of the form:

$$S(w) = \gamma \sum_{k=1}^{N_s} \left(y_{pred,k} - y_{true,k}\right)^2 + \zeta \sum_{p=1}^{N_w} w_p^2 \quad (10)$$

where $k = 1, \ldots, N_s$ represents the number of samples in the training dataset, $p = 1, \ldots, N_w$ represents the number of weights and biases in the neural network, y_{true} is the vector of target class label associated with each sample, y_{pred} is the vector of the predicted probabilities associated with each class label derived from the neural network, and γ and ζ are the regularization hyper-parameters. The minimization of $S(w)$ with respect to the weights and biases is a nonlinear optimization problem solved using the Levenberg–Marquardt algorithm, in which the gradient and the Hessian matrix of $S(w)$ are calculated using the backpropagation method. Assuming the weights and the data have Gaussian prior probability distributions, the regularization hyper-parameters, γ and ζ, are updated by maximizing their posterior probability distribution given the data, which is equivalent to maximizing the likelihood of evidence by Bayes' Theorem. Within each epoch, the cost function $S(w)$ is minimized with respect to w, and the likelihood of evidence is maximized with respect to γ and ζ. This is carried out iteratively until self-consistency is achieved, at which point the optimal distribution of weights and biases in the Bayesian regularized artificial neural network is obtained. Bayesian regularized artificial neural networks can effectively avoid over-training and over-fitting. Evidence procedures provide an objective Bayesian criterion for early stopping and remove the need for lengthy cross validations. Furthermore, the less relevant weights are turned off during the training process and Bayesian regularization effectively prunes the network.[31] Training and testing accuracies are calculated using the ratio between number of correctly classified samples and total number of samples in the training and testing sets, respectively. To develop a neural network detection model, closed-loop measurement data, both y_c and y_a, under two-tier control are collected. For better detection accuracy, training data needs to be collected starting at a broad range of initial conditions within the stability region Ω_ρ, such that various state evolutions under different operating conditions are covered. Full state measurements are recorded along the time-varying trajectory, and the Lyapunov function $V(x)$ is computed. As it captures the dynamic features of all states, $V(x)$ is an effective one-dimensional input feature for the attack detection problem. To ensure training accuracy, equal number of samples within each class are collected, with each sample corresponding to a different set of initial conditions for the closed-loop system simulation.

After data collection and adequate training, the NN detector is implemented online with the process controlled by the two-tier control system. The feed-forward NN model is a static model receiving inputs of fixed dimension, N_T, which is the length of the time-varying trajectory. Therefore, the detection window of the NN detector while implemented online also matches the trajectory length of the training data, N_T. The detector is activated every time full state measurements become available, and uses a moving horizon detection window, receiving latest sequences of $x(t_k)$ of fixed length N_T. Moreover, as the NN detector does not have perfect classification accuracy, false alarms may occur where large oscillatory data within normal ranges may be misclassified as a cyber-attack. To reduce false alarm rates, a sliding alarm verification window is also implemented, where the number of positive attack detections within this window need to surpass a threshold before a cyber-attack alarm is confirmed. The size of this verification window and the threshold value are determined based on the closed-loop evolution of the process, as these two parameters have a direct impact on the detection time and alarm rate. If sensor isolation is required, then all upper-tier state trajectories need to be fed into the neural network individually, as the output class labels depend on changes in each sensor. Each sample consists of a two-dimensional matrix $n_x \times N_T$, where n_x is the full state dimension, and N_T is the length of each state trajectory within the training simulation period. Similarly, equal number of samples in each class (i.e., one class representing each networked sensor measurement being attacked) are collected for various initial conditions in the stability region. These samples are used to train a sensor-isolation NN algorithm outputting multiple classes, where each class corresponds to each of the networked sensors being attacked. During online implementation, given that the system is under attack, this sensor isolator examines all states in the most recent N_T sampling periods and outputs which sensor is experiencing abnormalities.

Remark 2 *In the sliding verification window, we examine the number of positive detections out of the total number of detector activations; the two parameters, size of verification window and threshold for alarms, are different from a threshold number that is often examined in statistical methods such as Cumulative Sum. Long-term attacks such as geometric and surge attacks may be designed such that the cumulative error of the attacked measurements stay just below the statistical detection threshold, thus remain undetectable. However, they are detectable by neural network detectors, given that the extent and pattern of the attacked measurements are similar to the anomalous behavior learned by the neural net during training. Furthermore, neural network detectors trained with noisy sensor data are able to differentiate cyber-attacks from normal device fluctuations. However, in the case that measurement noise is so significant that it is similar to attacked oscillations (like in a replay attack), then the neural network detector may flag these noisy measurements as replay cyber-attacks. If significant noise is bound to be observed, then new neural network detectors can be readily trained based on these new noisy data to reflect the changed nominal operating conditions.*

4.4 | Mitigation measures via control system reconfiguration

Upon detection of an attack on the sensors providing networked asynchronous state measurements to the two-tier control system, the

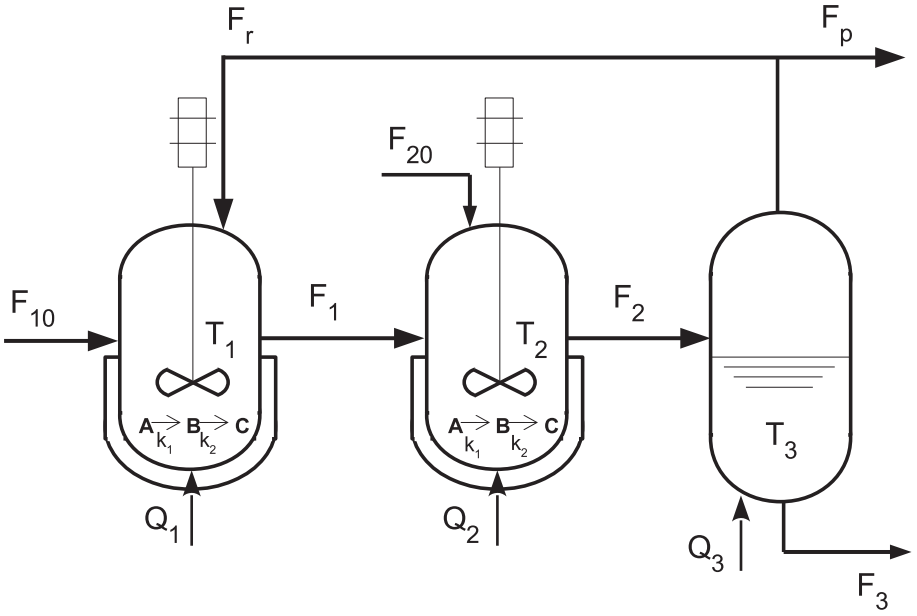

FIGURE 2 Process schematic consisting of two CSTRs and a flash drum separator

control system reconfiguration logic allows for two mitigation plans. First, the control system can deactivate the upper-tier controller completely and operate the system under the stabilizing lower-tier control system only, which uses cyber-secure, dedicated sensor measurements. Since the lower-tier controllers are capable of driving the process to its operating steady state with secure continuous measurements, the effect of the cyber-attacks is fully eliminated in the closed-loop system in this case and the process is stabilized to the operating steady state. Second, if a sensor isolation detector is also implemented, it will be activated once a sensor attack is verified. Subsequently, the upper-tier controller can choose to switch the compromised sensor to its redundant back-up sensor with secure readings. By abandoning the corrupted sensor and using its back-up sensor using a secure sensor-controller communication, the upper-tier controller remains functional and is able to drive the process to its steady state with better closed-loop performance.

In the extreme case that both continuous and asynchronous sensor measurements are attacked, the upper-tier controller will be shut off and the lower-tier controllers will reroute their continuous measurement signals from the corrupted sensors to their respective secure back-up sensors. The robustness of the proposed two-tier control architecture against intelligent cyber-attacks is demonstrated in Section 5 below through a reactor-reactor-separator process.

5 | APPLICATION TO A REACTOR-REACTOR-SEPARATOR PROCESS

5.1 | Process description and control system design

To simulate a chemical process application where multiple manipulated inputs are regulated by both the upper-tier and lower-tier controllers, a process network consisting of two continuous stirred tank reactors (CSTRs) followed by a flash tank separator is considered.[32] A schematic diagram of this process network can be found in Figure 2. Two reactions in series take place (A → B → C) in both reactors, and the overhead vapor from the flash tank is recycled to the first CSTR. All three vessels are assumed to have constant holdup. Using mass and energy balances, the process model can be obtained, which includes nine nonlinear ordinary differential equations as shown below:

$$\frac{dx_{A1}}{dt} = \frac{F_{10}}{V_1}(x_{A10} - x_{A1}) + \frac{F_r}{V_1}(x_{Ar} - x_{A1}) - k_1 e^{\frac{-E_1}{RT_1}} x_{A1} \tag{11a}$$

$$\frac{dx_{B1}}{dt} = \frac{F_{10}}{V_1}(x_{B10} - x_{B1}) + \frac{F_r}{V_1}(x_{Br} - x_{B1}) + k_1 e^{\frac{-E_1}{RT_1}} x_{A1} - k_2 e^{\frac{-E_2}{RT_1}} x_{B1} \tag{11b}$$

$$\begin{aligned}\frac{dT_1}{dt} = {} & \frac{F_{10}}{V_1}(T_{10} - T_1) + \frac{(-\Delta H_1)}{\rho C_p} C_M k_1 e^{\frac{-E_1}{RT_1}} x_{A1} \\ & + \frac{(-\Delta H_2)}{\rho C_p} C_M k_2 e^{\frac{-E_2}{RT_1}} x_{B1} + \frac{Q_1}{\rho C_p V_1} + \frac{F_r}{V_1}(T_3 - T_1)\end{aligned} \tag{11c}$$

$$\frac{dx_{A2}}{dt} = \frac{F_1}{V_2}(x_{A1} - x_{A2}) + \frac{F_{20}}{V_2}(x_{A20} - x_{A2}) - k_1 e^{\frac{-E_1}{RT_2}} x_{A2} \tag{11d}$$

$$\frac{dx_{B2}}{dt} = \frac{F_1}{V_2}(x_{B1} - x_{B2}) + \frac{F_{20}}{V_2}(x_{B20} - x_{B2}) + k_1 e^{\frac{-E_1}{RT_2}} x_{A2} - k_2 e^{\frac{-E_2}{RT_2}} x_{B2} \tag{11e}$$

$$\begin{aligned}\frac{dT_2}{dt} = {} & \frac{F_{20}}{V_2}(T_{20} - T_2) + \frac{(-\Delta H_1)}{\rho C_p} C_M k_1 e^{\frac{-E_1}{RT_2}} x_{A2} \\ & + \frac{(-\Delta H_2)}{\rho C_p} C_M k_2 e^{\frac{-E_2}{RT_2}} x_{B2} + \frac{Q_2}{\rho C_p V_2} + \frac{F_1}{V_2}(T_1 - T_2)\end{aligned} \tag{11f}$$

$$\frac{dx_{A3}}{dt} = \frac{F_2}{V_3}(x_{A2} - x_{A3}) - \frac{F_r + F_p}{V_3}(x_{Ar} - x_{A3}) \tag{11g}$$

$$\frac{dx_{B3}}{dt} = \frac{F_2}{V_3}(x_{B2} - x_{B3}) - \frac{F_r + F_p}{V_3}(x_{Br} - x_{B3}) \tag{11h}$$

TABLE 1 Values and descriptions of process parameters and steady states of state and input variables

Parameter/value	Description
F_{10} = 5.04 m^3/hr	Feed flow rate of CSTR 1
F_r = 50.4 m^3/hr	Recycle stream flow rate
F_p = 5.04 m^3/hr	Purge stream flow rate
T_{10} = 300 K, T_{20} = 300 K	Feed temperatures of CSTR 1 & 2
V_1 = 1.0 m^3, V_2 = 0.5 m^3, V_3 = 1.0 m^3	Volume of 3 vessels
$k_1 = 9.972 \times 10^6$ h^{-1},	Pre-exponential factors for reactions 1 & 2
$k_2 = 9.36 \times 10^6$ h^{-1}	
$E_1 = 5.0 \times 10^4$ kJ/kmol	Activation energy for reactions 1 & 2
$E_2 = 6.0 \times 10^4$ kJ/kmol	
$\Delta H_1 = -1.2 \times 10^5$ kJ/kmol	Heat of reaction for reactions 1 & 2
$\Delta H_2 = -1.4 \times 10^5$ kJ/kmol	
kJ/kmol $\Delta H_{vapA} = -3.53 \times 10^4$ kJ/kmol	Heat of vaporization for A, B, C
$\Delta H_{vapB} = -1.57 \times 10^4$ kJ/kmol	
$\Delta H_{vapC} = -4.068 \times 10^4$ kJ/kmol	
C_p = 4.2 kJ/(kg K)	Heat capacity
R = 8.314 kJ/(kmol K)	Gas constant
ρ = 1,000 kg/m^3	Liquid solution density
α_A = 3.5, α_B = 1.0, α_C = 0.5	Relative volatility of A, B, C
C_M = 2 kmol/m^3	Total molar concentration
x_{A1s} = 0.1762, x_{A2s} = 0.1965, x_{A3s} = 0.0651	Steady-state values of state variables
x_{B1s} = 0.6731, x_{B2s} = 0.6536, x_{B3s} = 0.6703	
T_{1s} = 480.32 K, T_{2s} = 472.79 K, T_{3s} = 474.89 K	
$Q_{1s} = 2.9 \times 10^9$ kJ/hr, $Q_{2s} = 1.9 \times 10^9$ kJ/hr,	Steady-state values of input variables
$Q_{3s} = 2.9 \times 10^9$ kJ/hr, F_{20s} = 5.04 m^3/hr	

$$\frac{dT_3}{dt} = \frac{F_2}{V_3}(T_2 - T_3) + \frac{Q_3}{\rho C_p V_3} + \frac{(F_r + F_p)C_M}{\rho C_p V_3}(x_{Ar}\Delta H_{vapA} + x_{Br}\Delta H_{vapB} + x_{Cr}\Delta H_{vapC}) \tag{11i}$$

where the state variables include the temperatures of the three vessels T_1, T_2, T_3, respectively, which are measured securely and continuously, and the mass fractions of species A and B in the three vessels x_{A1}, x_{A2}, x_{A3} and x_{B1}, x_{B2}, x_{B3}, whose measurements are available at asynchronous time instants and are sent to the upper-tier control system over a digital network that may be subjected to cyber-attacks. The upper-tier control system involves an LMPC that receives both asynchronous and continuous state measurements, and it is executed when full state information becomes available. Each of the three vessels has an external heat input. Three PI controllers are used to manipulate the heat inputs to the three vessels, Q_1, Q_2, and Q_3, each to regulate vessel temperature at a desired set-point value, and the LMPC manipulates the feed stream flow rate to second CSTR, F_{20}, to improve the speed of the closed-loop response. Assuming that there is negligible reaction in the separator tank and the relative volatility of each species remains constant within the operating temperature range, the composition of the recycle stream are: $x_{Ar} = \frac{\alpha_A x_{A3}}{\alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3}}, x_{Br} = \frac{\alpha_B x_{B3}}{\alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3}}, x_{Ar} = \frac{\alpha_C x_{C3}}{\alpha_A x_{A3} + \alpha_B x_{B3} + \alpha_C x_{C3}}$, where α represents the constant relative volatility of each species. Each of the six mass fraction measurements can be subject to the cyber-attacks, which are designed based on the current value of the true states at the time the attack occurs, as discussed in Section 4. With the integration of a machine-learning-based cyber-attack detector, the control objective is to track all nine states to an unstable equilibrium point while meeting all imposed constraints and staying immune to intelligent cyber-attacks. All process parameter values, the steady-state values, and the corresponding steady-state input values are given in Table 1. Deviation variables are used to present the simulation results, where the state vector and the input vector are represented as the difference between their values and their steady states. By using deviation variables, the equilibrium point of the process (i.e., the operating steady state) is at the origin of the state space. The input variables in deviation variable form are subject to the following operating constraints: -4.04 m^3/hr $\le \Delta F_{20} \le 3.96$ m^3/hr, $|\Delta Q_1| \le 5 \times 10^7$ kJ/hr, $|\Delta Q_2| \le 5 \times 10^7$ kJ/hr, $|\Delta Q_3| \le 5 \times 10^7$ kJ/hr.

Classical controllers are used in the lower-tier control system; specifically, proportional-integral (PI) controllers are used. The formulation of PI controller is presented as below:

$$u_{c_i}(t) = K_{c_i}\left(e_{c_i}(t) + \frac{1}{\tau_i}\int_0^t e_{c_i}(\tau)d\tau\right),\ e_{c_i}(t) = y_{c_i}^{REF}(t) - y_{c_i}(t) \tag{12a}$$

where $e_{c_i}(t)$ is the error between the measured output values y_{c_i} and their operating set-points $y_{c_i}^{REF}$ (defined based on the operating steady state), and K_{c_i} and τ_i are the proportional gain and integral time constant of each PI controller i = 1, 2, 3, respectively. In order to ensure closed-loop stability under PI control, K_{c_i} and τ_i are selected by first linearizing the model in Equation (1) around the steady state, and then assessing the eigenvalues of the linearized model $\dot{x} = Ax + Bu_c$. The proportional gain and time constant of the three PI controllers are chosen to be $[K_{c_1}\ K_{c_2}\ K_{c_3}]^T = \left[-8 \times 10^5, -8 \times 10^5, -8 \times 10^5\right]^T$ and $[\tau_1\ \tau_2\ \tau_3]^T = [5{,}000, 5{,}000, 5{,}000]^T$, respectively. An initial set of the PI controller parameters are determined using the Cohen-Coon tuning method, and then further optimized from closed-loop simulations, to make sure that the closed-loop response is smooth with reasonable control actions. With these tuning parameters, closed-loop stability under P-only control is ensured as the eigenvalues of the linearized model are Λ = [−2.599, −56.97, $-99.98 - 26.28i$, $-99.98 + 26.28i$, $-27.93 - 149.2i$, $-27.93 + 149.2i$, $-257.8 - 26.93i$, $-257.8 + 26.93i$, −758.8], all of which having negative real parts, and the integral term

aims to eliminate the offset while having minimal impact on the control action. An anti-windup mechanism is also implemented inside each PI controller to avoid integral wind-up effects which involves eliminating the integral term when the control action hits constraints. The upper-tier LMPC used in this simulation adopts the formulation shown in Equation (3). The objective function used in the optimization problem of LMPC is defined by a positive definite function, $L(x,u_a) = x^T Q_c x + u_a^T R_c u_a$, where R_c and Q_c are weighting matrices to penalize u_a and x, and have the following values: R_c = 1.0 and Q_c = *diag*([5,000, 10, 0.001, 5,000, 10, 0.001, 5,000, 10, 0.001]). The quadratic control Lyapunov function used in the contractive constraints of LMPC has the form $V(x) = x^T P x$, where P is a positive definite matrix: P = *diag*([3,228.31, 220.79, 4.334×10^{-4}, 2,576.72, 233.80, 4.474×10^{-4}, 23,675.92, 222.77, 4.434×10^{-4}]). The family of piece-wise constant function $S(\Delta)$ which u_a belongs to uses a sampling period of Δ = 0.02 hr, and the prediction horizon of the LMPC is N = 10. The nonlinear optimization problem of LMPC is solved using the OPTI-Toolbox in MATLAB. To numerically simulate the dynamic process model in Equation (11), explicit Euler method is used with an integration step of $h_c = 10^{-4}$ hr. The time sequence at which asynchronous measurements are sampled and received by the upper-tier controller is modeled after a lower-bounded random Poisson process, with each unequal interval between two consecutive asynchronous measurements being at least $\Delta_{a_k} \geq \Delta$ for all $k \in [1, N_T]$. The sequence of asynchronous intervals used in this simulation in which the LMPC calculations are executed is as follows: Δ_a = [0.04, 0.08, 0.1, 0.06, 0.12, 0.08, 0.02] for every 1.5 hr; alternative calculations of the asynchronous time instants may be considered with similar conclusions. After a simulation grid search, we use ρ = 120 as a level set of Lyapunov function to characterize the stability region and $\rho_{\min}$ = 0.1 to ensure convergence close to the steady state. The safe operating envelope of the 9 states in deviation variable form is as follows: x_l = [−0.1763, −0.6731, −50, −0.1965, −0.6536, −50, −0.0651, −0.6703, −50]^T denotes the lower bounds of the states and x_u = [0.7237, 0.2269, 50, 0.7035, 0.2464, 50, 0.8349, 0.2297, 50]^T denotes the upper bounds of the states. The stability region and the operating envelope are key parameters to generating intelligent cyber-attacks. The simulation period used for collecting training data is 3 hr, within which the lower-tier PI controllers are executed 150 times, and the upper-tier LMPC is executed 42 times. With the upper-tier controller receiving full-state measurements 42 times, the time-varying trajectories of state measurements have a length of

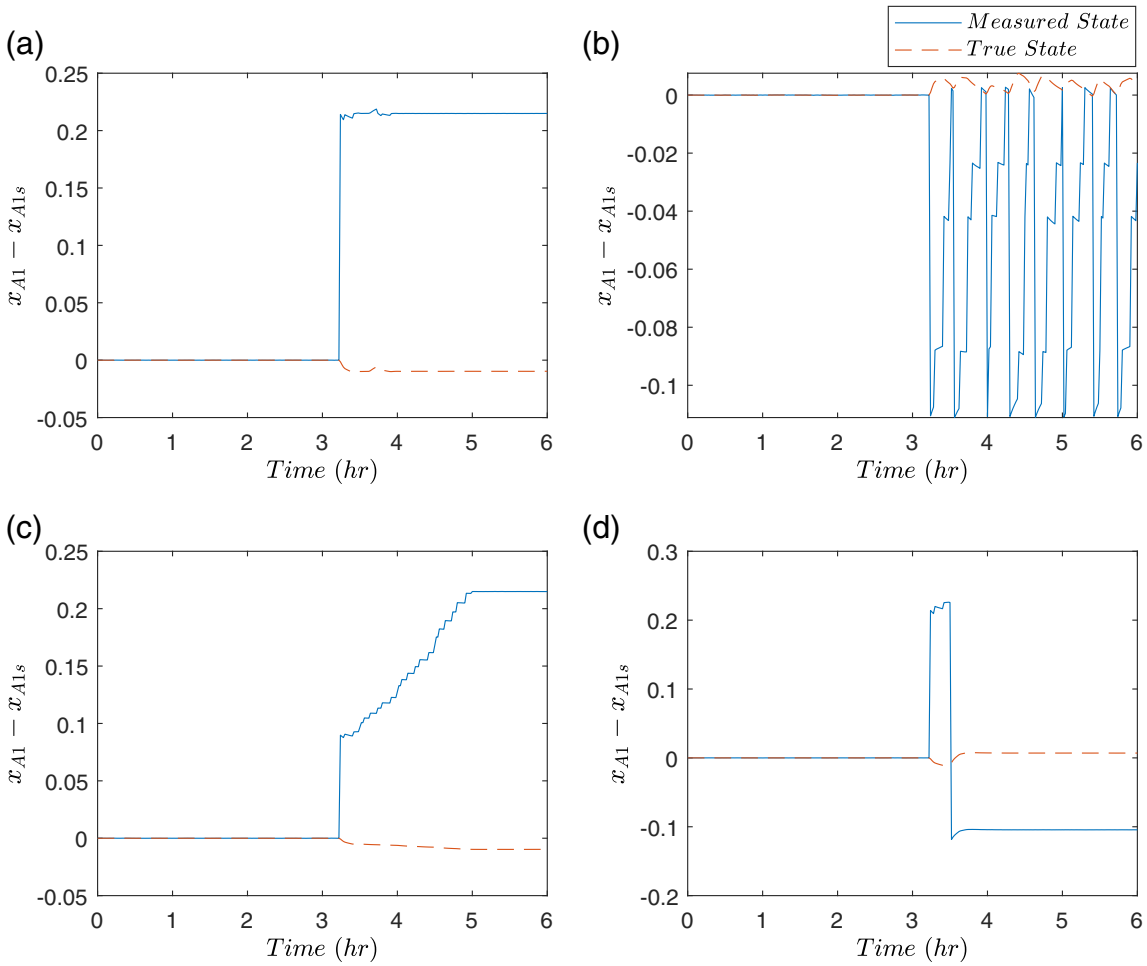

FIGURE 3 True and measured values of x_{A1} in deviation variable form without detection or mitigation mechanisms when (a) min-max, (b) replay, (c) geometric, and (d) surge cyber-attacks are introduced at 3.22 hr on the concentration sensor measuring x_{A1} [Color figure can be viewed at wileyonlinelibrary.com]

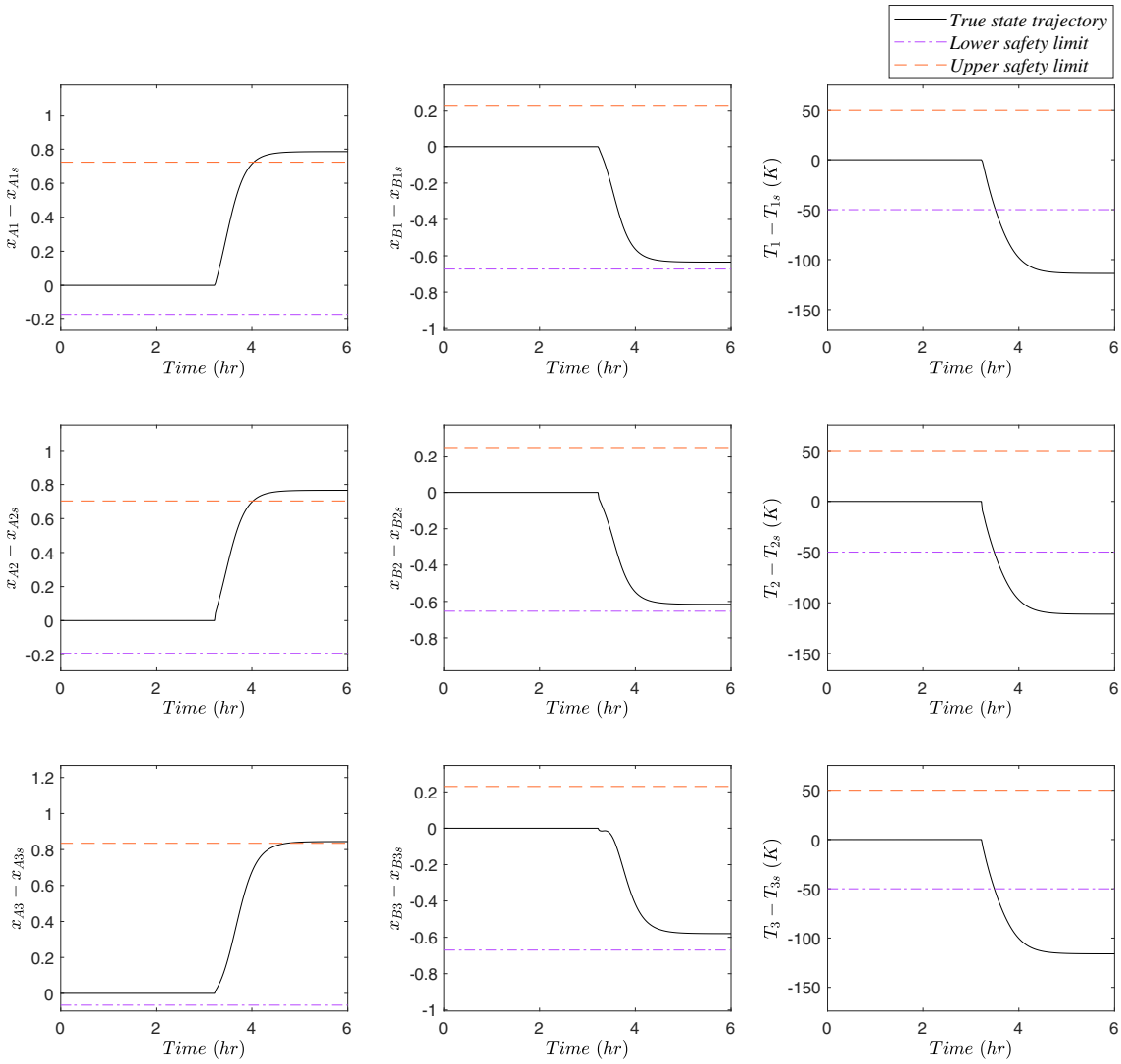

FIGURE 4 Evolution of true process states when min-max cyber-attacks on all nine state measurement sensors are introduced at 3.22 hr when the process network operates under the two-tier control architecture but no detection or control system reconfiguration mechanisms are implemented [Color figure can be viewed at wileyonlinelibrary.com]

N_T = 43, accounting for the initial condition measurements. Closed-loop simulations under two-tier and under PI-only control are carried out to compare the closed-loop performances; the initial conditions used to evaluate the performance metrics are x_0 = [0.0176, 0.067299, 48.032, 0.0197, 0.0654, 47.279, 0.006499, 0.067, 47.489]. Performance metrics in terms of settling time and normalized cumulative mean squared error along the state trajectories are calculated for closed-loop control under only lower-tier PI controllers, and under the two-tier LMPC/PI control scheme. It is shown that it takes 2.46 hr for lower-tier PI controllers, and 0.6 hr for two-tier LMPC/PI to settle to the operating steady state. The normalized cumulative mean squared errors are 4.1203 and 0.8014 for lower-tier PI and two-tier LMPC/PI, respectively. The two-tier control architecture achieves significantly better closed-loop performance by stabilizing the process within shorter time and eliminating process overshoots and offset effectively.

5.2 | Cyber-attacks and detector training

Min-max attacks are used to train the neural-network-based detector with and without sensor noise. If the neural network detector is trained with only one type of attack, the resulting output will have two classes—attacked and not attacked. In addition, replay attacks are also used to train a neural network detector capable of identifying the type of attack, where the output classes consist of three labels: not attacked, attacked by min-max attacks, and attacked by replay attacks. In the first five sampling steps, more extreme oscillations with larger magnitudes in state feedback are observed. Therefore, these aggressively oscillatory measurements with length L_a = 5 are recorded and used as replay attacks. Other attacks with varying lengths can be introduced at random time instants between $i_0 \in [6, 42]$ to simulate cyber-attacks of various durations and occurring at various times during operation. With extensive closed-loop simulations, equal number of samples are collected for each output class,

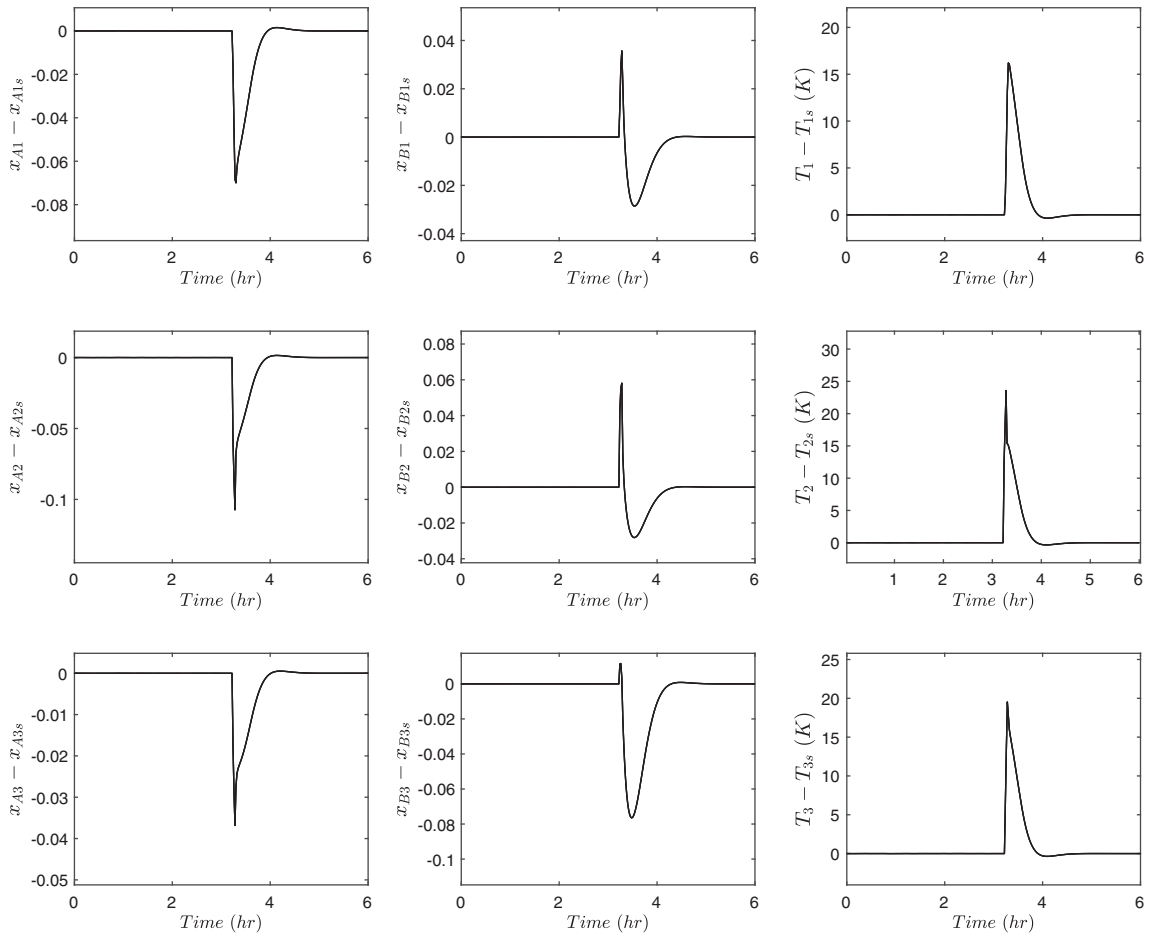

FIGURE 5 Evolution of true process states under min-max cyber-attacks on all nine state measurements. The min-max cyber-attacks are introduced at 3.22 hr and are detected at 3.28 hr, at which time the upper-tier LMPC is turned off and the temperature measurements used by the lower-tier PI controllers are taken from secure back-up temperature sensors and the process is driven back to the steady state

with each sample either consisting of a 1×43 array of V (x) values (in attack identification), or a 9×43 matrix of x values along the dynamic trajectory (in sensor isolation), where the 9×43 matrix in each sample is then collapsed into a 1×387 array to be fed into the feed-forward neural network detector. Four NN detectors are trained to carry out detection: (a) a 2-class model with nominal operation under min-max attack (12,000 samples per class label, training time 24.05 s), (b) a 2-class model with noisy sensors under min-max attack (1,044 samples per class label, training time 4.332 s), (c) a 3-class model with noisy sensors and 2 attack types—min-max and replay (1,044 samples per class label, training time 5.265 s), and (d) a 6-class model with noisy sensors under min-max attack for corrupted sensor isolation (2,800 samples per class label, training time 6,211.52 s). To train the compromised sensor isolation detector, we also used a min-max attack to simulate the abnormal behaviors. Noisy sensors are simulated by adding bounded Gaussian white noise on each sensor. The lower and upper bounds of the sensor noises are as follows: $|w_1| \le 7.5 \times 10^{-5}$, $|w_2| \le 5.5 \times 10^{-5}$, $|w_3| \le 0.032$ K, $|w_4| \le 7.5 \times 10^{-5}$, $|w_5| \le 5.5 \times 10^{-5}$, $|w_6| \le 0.032$ K, $|w_7| \le 3.5 \times 10^{-5}$, $|w_8| \le 5.5 \times 10^{-5}$, $|w_9| \le 0.032$ K. These Gaussian noise distributions have a mean of $\mu = 0$ and standard deviations $\sigma_1 = \sigma_4 = 0.0002$, $\sigma_2 = \sigma_5 = \sigma_8 = 0.001$, $\sigma_3 = \sigma_6 = \sigma_9 = 0.1$ K, and $\sigma_7 = 0.0001$. Feed-forward neural networks with two hidden layers having 12 and 10 neurons, respectively, are built using the MATLAB Machine Learning and Deep Learning Toolboxes. Both hidden layers use a *tansig* activation function, which is in the form $g_{1,2}(z) = \frac{2}{1+e^{-2z}} - 1$, and is commonly known as the hyperbolic tangent sigmoid transfer function. The output layer uses a *softmax* function to provide a predicted probability of the class labels, which is in the form of $g_3(z_j) = \frac{e^{z_j}}{\sum_{i=1}^{H} e^{z_i}}$ where H denotes the number of class labels. The NN detector trained with nominal conditions has a training accuracy of 99.6% and a testing accuracy of 92.2%, while the NN detector trained with noisy sensors achieves a training accuracy of 99.9% and testing accuracy of 100%. The NN algorithm trained with noisy sensors achieves a higher accuracy than the nominal case because the addition of noise contributes more variance to the training dataset, thereby making the learning process harder and yielding a more robust NN detector. Moreover, the training and testing accuracy of the NN detector trained with noisy sensors under two types of cyber-attacks are 98.2% and 91.4%, respectively, and the NN algorithm to isolate the compromised noisy sensor achieves a training and testing accuracy of 99.6% and 99.0%, respectively.

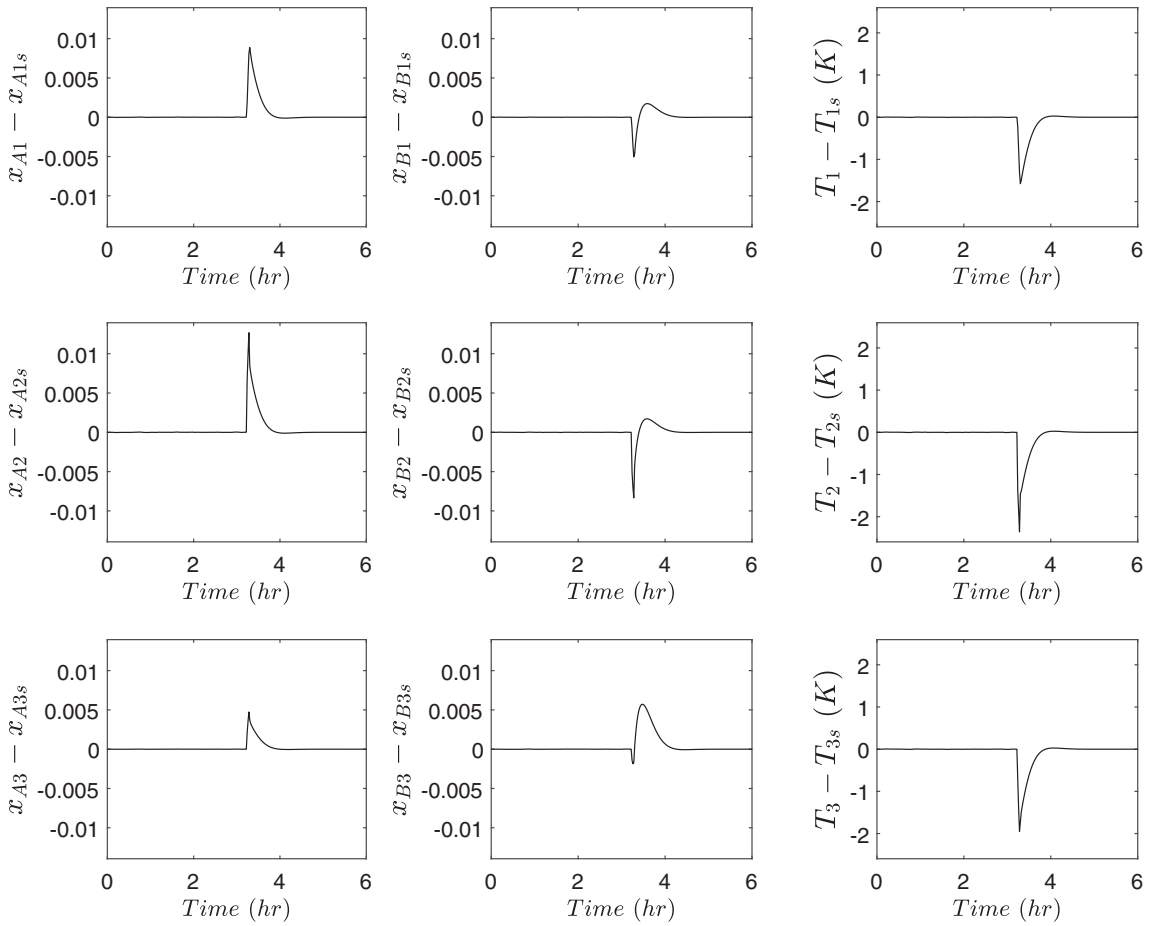

FIGURE 6 Evolution of true process states under min-max cyber-attacks on all six mass fraction sensors. The min-max cyber-attacks are introduced at 3.22 hr and are detected at 3.28 hr, at which time the upper-tier LMPC is turned off and the process is driven back to the steady state under the lower-tier PI controllers

5.3 | Cyber-attack detection results

The NN detectors are implemented online with the process operated under two-tier control with initial conditions at the operating steady state; i.e., $x_0 = [0, 0, 0, 0, 0, 0, 0, 0, 0]^T$ (the conclusions are similar for other initial conditions in Ω_ρ). Therefore, attacks are introduced when the process is stabilized at its operating steady state. As the NN detectors are trained using a fixed input dimension of 43 (with 42 sampling steps), to ensure that input data with sufficient length is collected, the detector is activated at time instant k = 42 in the asynchronous time sequence, which corresponds to real time t = 3.0 hr. The detector reads state measurements in the previous 42 sampling periods, analyzes their behaviors, and computes an output on which class these time-series data resembles. The window of this fixed-length segment of time-sequence data rolls forward in time as the upper-tier LMPC and the attack detector are executed in real time. The alarm verification window is chosen to be three sampling periods of the upper-tier LMPC, where two positive detections within every three consecutive sampling steps will confirm the presence of an attack. Once an attack is confirmed, at the same time instant, the detection alarm will be triggered and the LMPC will be deactivated. Furthermore, to examine whether the detector will misclassify not-attacked signals as being under attack, attacks are introduced a few sampling periods after the detector has been activated at t = 3.0 hr, such that the first few outputs by the detector are based on normal operation data. Cyber-attacks with a duration of L_a = 40 sampling periods are introduced at time instant i_0 = 45, which corresponds to t = 3.22 hr; thus, the compromised sensor will stay corrupted until the end of the 6-hr simulation period. To illustrate the pattern and effect of the four cyber-attack types, Figure 3 shows the true state values and the sensor values of state 1 when min-max, replay, geometric, and surge attacks target only the sensor measuring mass fraction x_{A1} with bounded noise. Although Figure 3 only shows the true state progression of state 1, all nine states experience similar deviating patterns after the cyber-attacks. Under min-max attack, the true state settles at an offset of similar magnitude as the initial jump. Replay attack results in aggressive oscillations in true plant states around an offset. Geometric attacks drive process states increasingly away from the operating steady state before reaching an offset due to the increasing magnitude of the attack with time until the attack reaches the boundary of the stability region. Surge attack causes an initial jump similarly seen in min-max attacks; with the reduction of attack severity, states

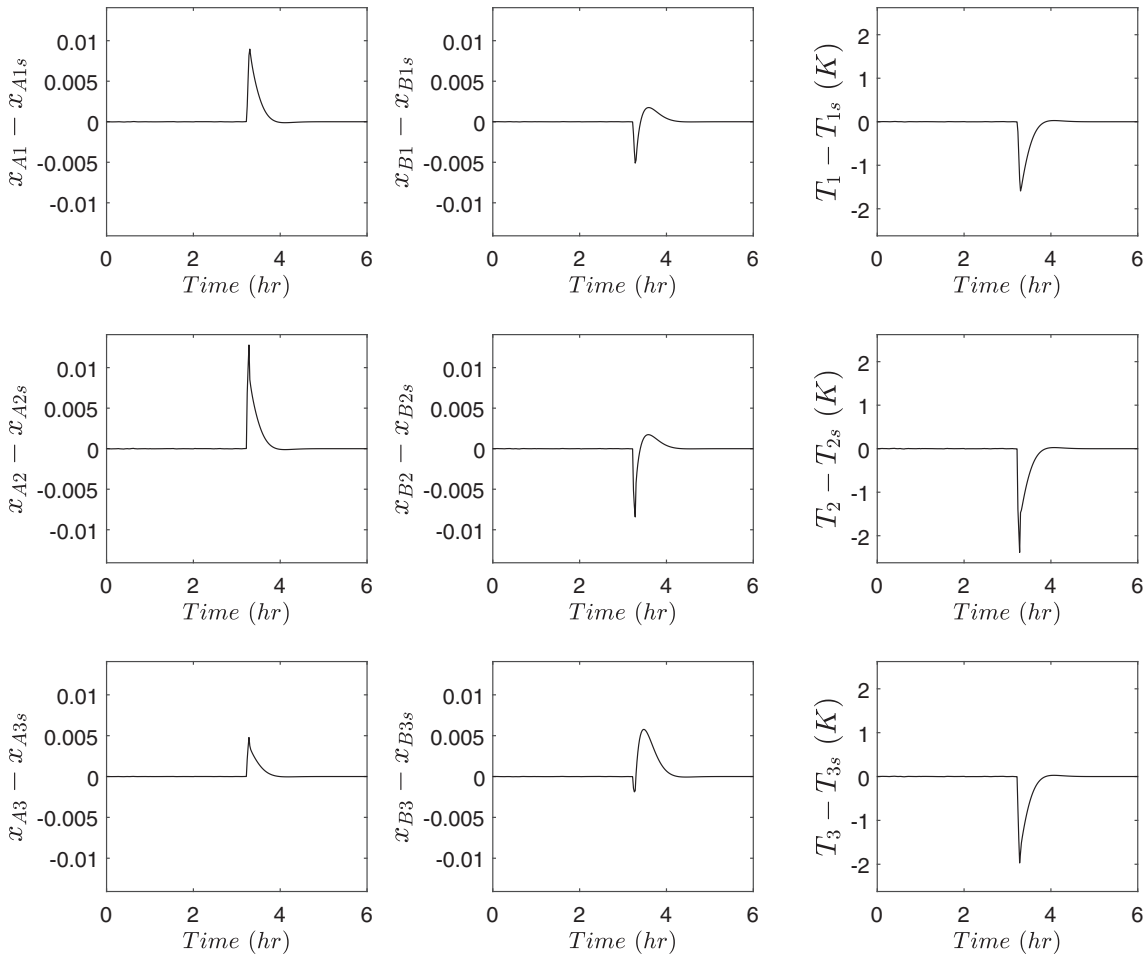

FIGURE 7 Evolution of true process states under replay cyber-attacks on all six mass fraction sensors. The replay cyber-attacks are introduced at 3.22 hr and are detected at 3.28 hr, at which time the upper-tier LMPC is turned off and the process is driven back to the steady state under the lower-tier PI controllers

are driven closer to the setpoint, but still reach an offset that is smaller than that in min-max. The upper-tier LMPC receives falsified information on the values of these process states, and in turn, computes a control action that is unable to drive the true states back to the steady state. States do not continue to diverge and do not exit the stability region, but instead settle at an offset value, due to the stabilizing contributions from the three lower-tier PI controllers, which use secure sensor measurements. Regardless, the attack has successfully targeted the system and the two-tier control system fails to drive the states back to their operating steady states without using any data-based detection algorithms.

When the trained NN detectors are applied during online operation, time delays are observed due to the configured alarm verification window, which requires at least two positive detections out of three detection instances to confirm an attack. The time delay is defined as the number of sampling steps between when the attack is inserted and when the attack is confirmed. In the cases of the first two detectors trained using min-max attack only (i.e., nominal and with noise operations), replay, geometric and surge attacks are unknown attacks which have not been learned by the NN detector. All four types of cyber-attacks, despite the latter three being unknown to the NN detector, are captured by the NN detector trained under nominal condition. The NN detection algorithm detects min-max, replay, and surge attacks successfully with a time delay of one sampling period. This is because two out of three detections need to be positive to confirm a detection; in other words, as soon as two consecutive positive detections occur, the detection is confirmed. Therefore, the detection of these cyber-attacks is delayed by one sampling period, at which time the second consecutive positive detection is received by the control system. A time delay of two sampling periods is observed when a geometric attack is introduced due to the initial small magnitude of change induced, therefore causing the NN detector a delay in predicting the correct class label. As time progresses, the attack increases exponentially towards a point where the deviation is on par with the other three attacks, at which point the detector captures the irregular deviation. The potential time delay of NN detectors trained with min-max attacks in detecting geometric attacks will vary depending on the geometric parameters, that is, β and α in Equation (7). Meanwhile, the NN detector trained with noisy sensor measurements is able to detect min-max, geometric, and surge attacks successfully, but with a time delay of seven sampling periods when the geometric attack is introduced. Moreover, this detector fails to detect replay attacks due to the oscillatory nature of the replay signals. Unlike the other three attacks where the attacked measurement

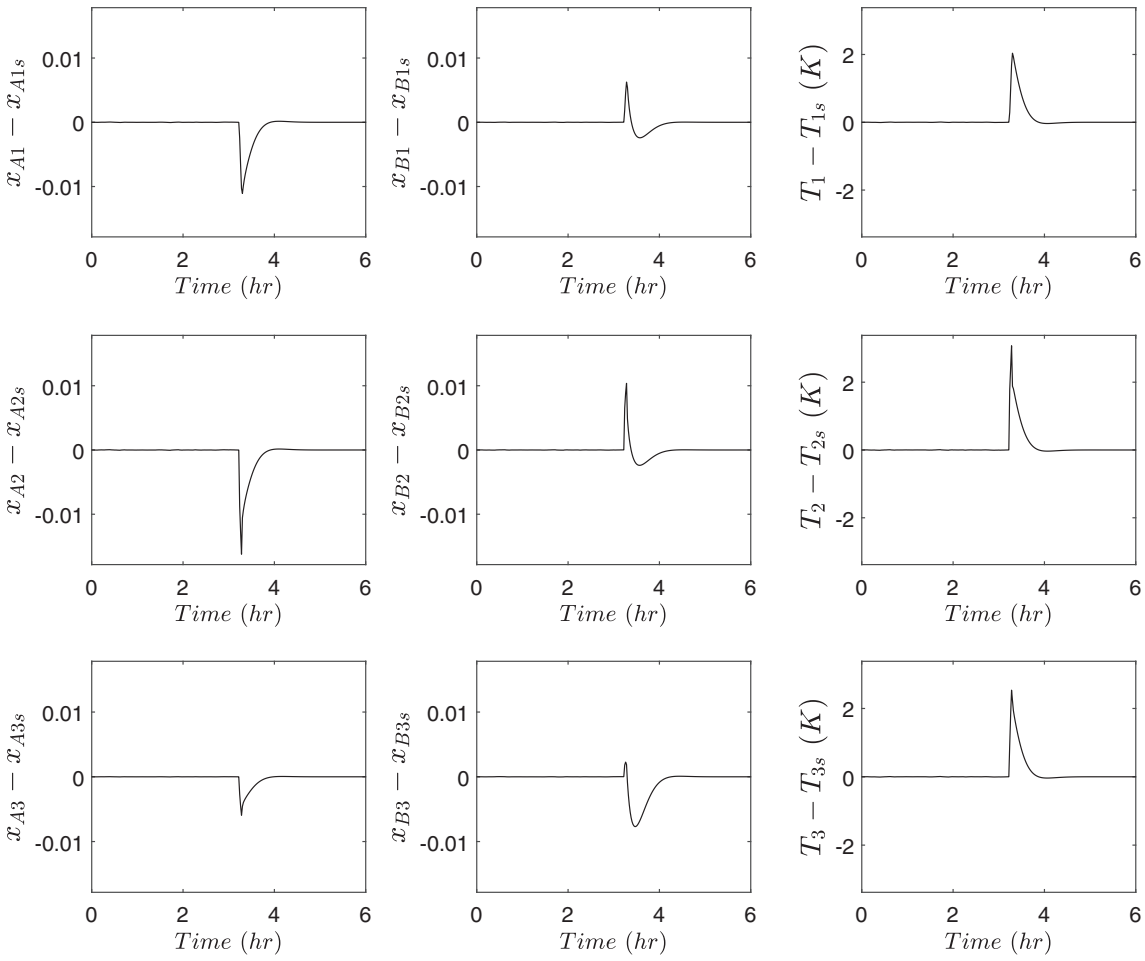

FIGURE 8 Evolution of true process states under geometric cyber-attacks on all six mass fraction sensors. The geometric cyber-attacks are introduced at 3.22 hr and are detected at 3.28 hr, at which time the upper-tier LMPC is turned off and the process is driven back to the steady state under the lower-tier PI controllers

stays at the attack target for at least two sampling periods, replay attacks oscillate at every sampling period, which is different from the min-max behavior that this NN detector is trained based on. Given the relatively smaller magnitude of the oscillations (not reaching the min-max window) and oscillatory behavior of replay attacks, the NN detector trained with added noises is not able to differentiate replay attacks from sensor noise. Thus, a third NN detector is trained, where both replay and min-max attacks are accounted in the training process. This detection algorithm outputs three classes, where min-max and replay attacks are classified correctly and the detection is confirmed after a time delay of one sampling period, and geometric and surge attacks are classified as replay attacks with the detection confirmed after one sampling period.

Cyber-attacks could target multiple sensors at once, and the detection algorithms are tested on these cases where more than one sensor could be under attack. We first consider the extreme case where min-max cyber-attacks are applied on all nine sensors to simulate the impact of cyber-attacks when all state measurements are compromised without using any online detectors; this scenario allows to demonstrate the value of the proposed two-tier control architecture. The true state trajectories are shown in Figure 4 where a min-max attack is introduced at 3.22 hr with a duration covering until the end of the simulation period. With the continuous temperature measurements also under cyber-attack, closed-loop stability under the lower-tier controllers is no longer achieved. As a result of the cyber-attack, the true state evolution exits the stability region when no detection algorithms are being used; moreover, the mass fractions of species A and the temperatures in all three vessels exceed their operating boundaries, violating the safety limits on their states in deviation variable form. Under the circumstance that continuous temperature measurements are jeopardized, the only cyber-attack countermeasure is to reroute measured temperature signals received by lower-tier controllers from the corrupted sensors to a new set of redundant sensors with secure readings. This extreme scenario demonstrates the severity of the destabilizing impacts of cyber-attacks to all sensors, and thus, the necessity of having secure and reliable feedback measurements for the lower-tier controllers in order to maintain the robustness of the overall control architecture.

To mitigate the impact of the cyber-attacks on all nine sensors, the neural-network detection algorithm trained with noisy measurements and two cyber-attack types is applied online. With the alarm verification window to reduce false alarms, the min-max attack is introduced at 3.22 hr and the detection is confirmed at 3.28 hr, from which point the upper-tier LMPC is turned off and the continuous temperature

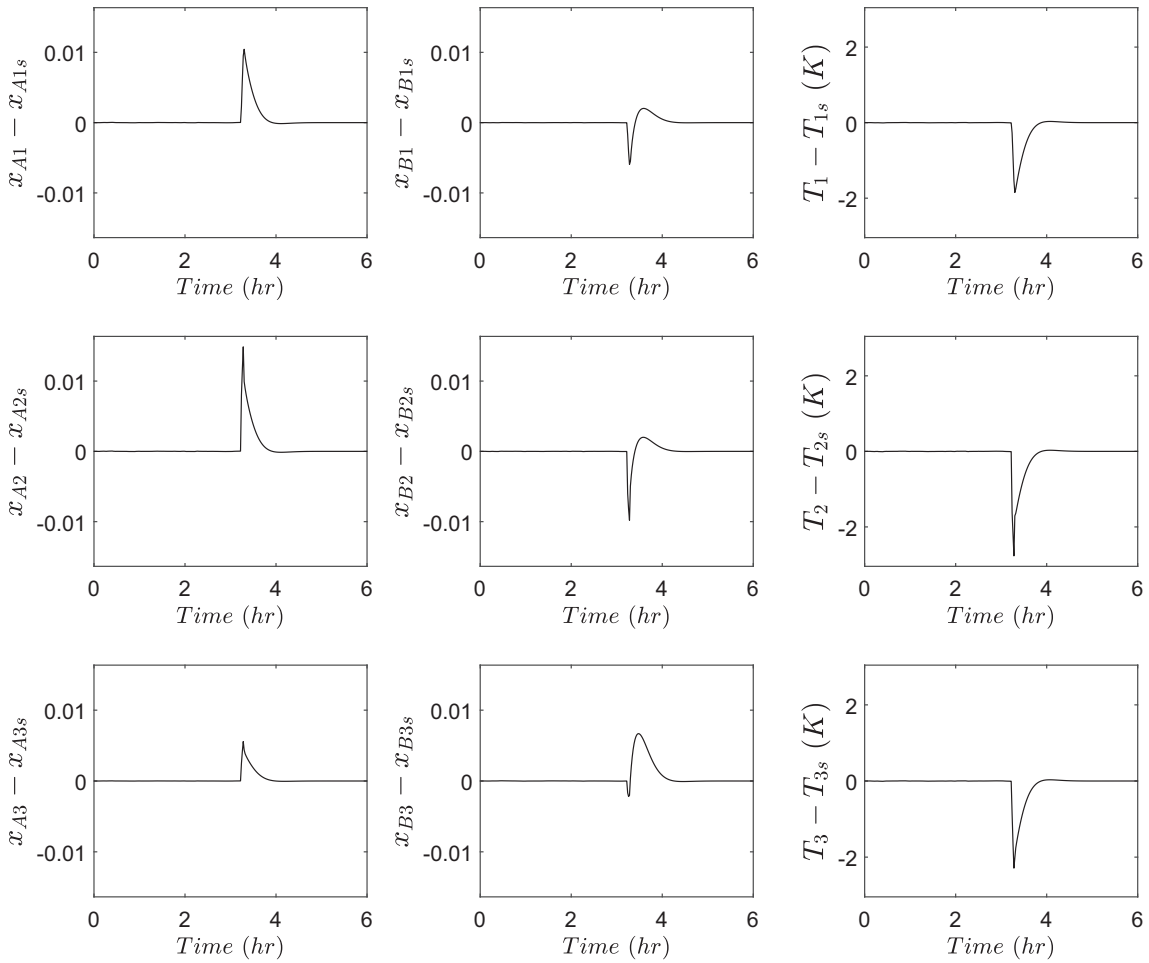

FIGURE 9 Evolution of true process states under surge cyber-attacks on all six mass fraction sensors. The surge cyber-attacks are introduced at 3.22 hr and are detected at 3.28 hr, at which time the upper-tier LMPC is turned off and the process is driven back to the steady state under the lower-tier PI controllers

measurements used by lower-tier PI controllers are obtained from a set of secure back-up sensors (Figure 5). By doing this, the closed-loop stability of the process under lower-tier control is re-established and the process is driven back to its operating steady state.

To ensure the robustness of the lower-tier controller against cyber-attacks, we now consider the case where only the networked mass fraction measurements fed into the upper-tier LMPC are attacked while the continuous temperature measurements used by both the lower-tier PIs and the upper-tier LMPC remain secure. Therefore, with the stabilizing lower-tier PI controllers, the upper-tier LMPC can be turned off once the detection of an attack is confirmed (i.e., u_a = 0 for the remainder of the closed-loop simulation) such that the false control actions calculated by the upper-tier LMPC will not act as a disturbance to the closed-loop system. The effectiveness of this mitigation strategy is illustrated in our simulation results, where the true plant states are driven back to their operating steady states using the stabilizing lower-tier PI controllers despite the sudden jumps or gradual deviations caused by cyber-attacks. The re-stabilized state trajectories after min-max, replay, geometric, and surge attacks are shown respectively in Figures 6–9. When the attacks are introduced at time 3.22 hr, the detection algorithm confirms that the measured state trajectory is under attack at 3.28 hr, at which point the LMPC is turned off, and the process is re-stabilized to its operating steady state using the lower-tier PI controllers. Despite the minor degradation in closed-loop performance with only lower-tier controllers, the reconfigured control system succeeds in maintaining closed-loop stability in the presence of cyber-attacks.

6 | CONCLUSION

In this work, a cyber-secure control architecture for nonlinear chemical processes incorporating secure lower-tier explicit feedback controllers and an upper-tier model predictive controller was proposed. On top of the stabilizing lower-tier controllers, the upper-tier LMPC contributed to better closed-loop performance by using networked sensor measurements which may be vulnerable to cyber-attacks. A neural-network-based detector is integrated with the two-tier control architecture such that the control system can be reconfigured to stabilize the process at the original steady state upon detection of a cyber-attack. Neural-network-based detection algorithms were developed and have

proven their success during online implementation in detecting the presence of cyber-attacks when various types of cyber-attacks were applied on multiple sensors. Four feed-forward neural networks were trained and tested under nominal and noisy operating conditions, all of which achieved a detection accuracy of over 91%. Through the application of the proposed detection and mitigation methods on a multivariable process, this work demonstrated the effectiveness of machine-learning-based methods in developing algorithms used for cyber-attack diagnosis and cyber-defense, as well as the robustness of the proposed two-tier control architecture in maintaining cyber-security.

ACKNOWLEDGMENT

Financial support from the National Science Foundation and the Department of Energy is gratefully acknowledged.

ORCID

Panagiotis D. Christofides https://orcid.org/0000-0002-8772-4348

REFERENCES

1. Christofides PD, Davis JF, El-Farra NH, Clark D, Harris KR, Gipson JN. Smart plant operations: vision, progress and challenges. *AIChE J*. 2007;53:2734-2741.
2. Stouffer KK, Falco J. *Recommended Practise: Improving Industrial Control System Cyber-security with Defense-in-Depth Strategies*. Control Systems Security Program, National Cyber Security Division: Department of Homeland Security; 2009.
3. Khorrami F, Krishnamurthy P, Karri R. Cybersecurity for control systems: a process-aware perspective. *IEEE Des Test*. 2016;33:75-83.
4. Polycarpou MM, Ioannou PA. *Identification and Control of Nonlinear Systems Using Neural Network Models: Design and Stability Analysis*. Technical report: University of Southern California; 1991.
5. Rawlings JB, Maravelias CT. Bringing new technologies and approaches to the operation and control of chemical process systems. *AIChE J*. 2019;65:e16615.
6. Wu Z, Tran A, Rincon D, Christofides PD. Machine learning-based predictive control of nonlinear processes. Part II: Computational implementation. *AIChE J*. 2019;65:e16734.
7. Wang H, Chaffart D, Ricardez-Sandoval L. Modelling and optimization of a pilot-scale entrained-flow gasifier using artificial neural networks. *Energy*. 2019;188:116076.
8. Chaffart D, Ricardez-Sandoval L. Optimization and control of a thin film growth process: a hybrid first principles/artificial neural network based multiscale modelling approach. *Comput Chem Eng*. 2018;119:465-479.
9. Kimaev G, Ricardez-Sandoval L. Nonlinear model predictive control of a multiscale thin film deposition process using artificial neural networks. *Chem Eng Sci*. 2019;207:1230-1245.
10. Bishop CM. *Pattern Recognition and Machine Learning*. New York, NY: Springer; 2006.
11. Murphey YL, Masrur MA, Chen ZH, Zhang B. Model-based fault diagnosis in electric drives using machine learning. *IEEE/ASME Trans Mechatron*. 2006;11:290-303.
12. West SR, Guo Y, Wang XR, Wall J. Automated fault detection and diagnosis of HVAC subsystems using statistical machine learning. In *Proceedings of the 12th International Conference of the International Building Performance Simulation Association*, Sydney, Australia, 2011.
13. Tsai CF, Hsu YF, Lin CY, Lin WY. Intrusion detection by machine learning: a review. *Expert Syst Appl*. 2009;36:11994-12000.
14. Buczak AL, Guven E. A survey of data mining and machine learning methods for cyber security intrusion detection. *IEEE Commun Surv Tutor*. 2015;18:1153-1176.
15. Ozay M, Esnaola I, Vural FTY, Kulkarni SR, Poor HV. Machine learning methods for attack detection in the smart grid. *IEEE Trans Neural Netw Learn Syst*. 2015;27:1773-1786.
16. Goh J, Adepu S, Tan M, Lee ZS. Anomaly detection in cyber-physical systems using recurrent neural networks. In *Proceedings of the 18th IEEE International Symposium on High Assurance Systems Engineering*, Singapore, 2017, pp. 140–145.
17. Hink RCB, Beaver JM, Buckner MA, Morris T, Adhikari U, Pan S. Machine learning for power system disturbance and cyber-attack discrimination. In *Proceedings of the 7th International Symposium on Resilient Control Systems*, Denver, CO, USA, 2014. IEEE, pp. 1–8.
18. Junejo KN, Goh J. Behaviour-based attack detection and classification in cyber physical systems using machine learning. In *Proceedings of the 2nd ACM International Workshop on Cyber-Physical System Security*, Xi'an, China, 2016, pp. 34–43.
19. Wu Z, Albalawi F, Zhang J, Zhang Z, Durand H, Christofides PD. Detecting and handling cyber-attacks in model predictive control of chemical processes. *Mathematics*. 2018;6:173.
20. Singh J, Nene MJ. A survey on machine learning techniques for intrusion detection systems. *Int J Adv Res Comput Commun Eng*. 2013;2:4349-4355.
21. Stewart BT, Venkat AN, Rawlings JB, Wright SJ, Pannocchia G. Cooperative distributed model predictive control. *Syst Control Lett*. 2010;59:460-469.
22. Pourkargar DB, Almansoori A, Daoutidis P. Comprehensive study of decomposition effects on distributed output tracking of an integrated process over a wide operating range. *Chem Eng Res Des*. 2018;134:553-563.
23. Yin X, Liu J. Subsystem decomposition of process networks for simultaneous distributed state estimation and control. *AIChE J*. 2019;65:904-914.
24. Mohanty SR, Pradhan AK, Routray A. A cumulative sum-based fault detector for power system relaying application. *IEEE Trans Power Deliv*. 2007;23:79-86.
25. Cárdenas AA, Amin S, Lin ZS, Huang YL, Huang CY, Sastry S. Attacks against process control systems: risk assessment, detection, and response. In *Proceedings of the ACM Symposium on Information, Computer and Communications Security*, Hong Kong, China, 2011, pp. 355–366.
26. Huang L, Nguyen X, Garofalakis MN, et al. Communication-efficient online detection of network-wide anomalies. In *INFOCOM*, volume 7, Anchorage, Alaska, 2007, pages 134–142.
27. Omar S, Ngadi A, Jebur HH. Machine learning techniques for anomaly detection: an overview. *Int J Comput Appl*. 2013;79:33-41.
28. Agrawal S, Agrawal J. Survey on anomaly detection using data mining techniques. *Proc Comput Sci*. 2015;60:708-713.
29. Gurney K. *An Introduction to Neural Networks*. Boca Raton, FL: CRC Press; 2014.
30. Sibi P, Allwyn JS, Siddarth P. Analysis of different activation functions using back propagation neural networks. *J Theor Appl Inf Technol*. 2013;47:1264-1268.
31. Burden F, Winkler D. Bayesian regularization of neural networks. *Artificial Neural Networks*. New York, NY: Springer; 2008:23-42.
32. Zhang J, Liu J. Distributed moving horizon state estimation for nonlinear systems with bounded uncertainties. *J Process Control*. 2013;23:1281-1295.

How to cite this article: Chen S, Wu Z, Christofides PD. A cyber-secure control-detector architecture for nonlinear processes. *AIChE J*. 2020;66:e16907. https://doi.org/10.1002/aic.16907