ASME Journal of Computing and Information Science in Engineering
Online journal at:
https://asmedigitalcollection.asme.org/computingengineering

GLHAD: A Group Lasso-Based Hybrid Attack Detection and Localization Framework for Multistage Manufacturing Systems

Ahmad Kokhahi
Department of Industrial Engineering,
Clemson University,
Freeman Hall,
Clemson, SC 29634
e-mail: akokhah@clemson.edu

Dan Li[1]
Assistant Professor
Department of Industrial Engineering,
Clemson University,
Freeman Hall,
Clemson, SC 29634
e-mail: dli4@clemson.edu

As Industry 4.0 and digitization continue to advance, the reliance on information technology increases, making the world more vulnerable to cyberattacks, especially cyber-physical attacks that can manipulate physical systems and compromise sensor data integrity. Detecting cyberattacks in multistage manufacturing systems (MMS) is crucial due to the growing sophistication of attacks and the complexity of MMS. Attacks can propagate throughout the system, affecting subsequent stages and making detection more challenging than in single-stage systems. Localization is also critical due to the complex interactions in MMS. To address these challenges, a group lasso regression-based framework is proposed to detect and localize attacks in MMS. The proposed algorithm outperforms traditional hypothesis testing-based methods in expected detection delay and localization accuracy, as demonstrated in a simple linear multistage manufacturing system. [DOI: 10.1115/1.4063862]

Keywords: artificial intelligence, cyber-physical security for factories, cybermanufacturing, machine learning for engineering applications

1 Introduction

Due to the advancement in automation and the industrial internet-of-things, concerns around the cybersecurity of manufacturers have grown considerably in recent years [1]. After the well-known Stuxnet [2] targeting Iran's nuclear program, cyberattacks aiming at disrupting manufacturing operations have surged. In 2022, the manufacturing industry became the top-target of cyberattacks in all operational technology (OT)-related industries. Cyberattacks affecting manufacturing companies are not new. For example, the WannaCry [3] ransomware affected several manufacturing companies, including Taiwan Semiconductor Manufacturing Company (TSMC). However, what is more alarming than attacks that aimed to simply shut down the system or bleach the data are the ones that aims to destroy the manufacturing assets or cause life-threatening events. In 2014, an attack against a German steel mill company [4] targeted the industrial control system and caused components of the plant controls to fail and caused the blast furnace to shut down improperly. In 2017, a malware was launched against a Saudi Arabian petrochemical plant and reprogrammed its safety systems which directly monitors and controls the equipment and processes. The above attacks are classified as cyber-physical attacks, where the attack intrudes into the system from the cyber network but aims to disrupt the physical process. Most cyber-physical attackers exploit the vulnerabilities in the design of supervisory control and data acquisition (SCADA) systems to design sophisticated sensor spoofing attacks based on techniques such as network traffic eavesdropping, traffic rerouting, IP spoofing, packet crafting, or session hijacking. Although secure communication protocols have been introduced for sensor data transmission, they do not prevent attacks injected to the sensors [5]. Further, due to the high volume of sensor data and limited computing and storage resources to encrypt or carry the encrypted data, current manufacturing operational (sensing and control) data are commonly being transmitted in an unencrypted manner, especially in legacy systems [6]. Further, smart attackers can leverage advanced techniques (such as machine learning) to expose the data or modify the data without decryption [7]. The above issues allow the attacker to manipulate the sensor data (compromising sensor data integrity) and lead to malicious control actions that will damage the manufacturing systems and the final products. In this paper, we focus on detecting and localizing sensor spoofing attacks at the level of manufacturing SCADA.

Multistage manufacturing systems play a crucial role in the manufacturing industry [8]. These systems, comprised of multiple components, stations, or stages, can be modeled as a series of interconnected elements working together to produce the final product [8,9]. The digital connectivity between components and devices in MMSs, as well as the standard data communication protocols used in manufacturing execution systems, make modern manufacturing systems vulnerable to cyberattacks. Additionally, due to the interconnectivity between stages in an MMS, a single

[1]Corresponding author.

Manuscript received May 31, 2023; final manuscript received October 12, 2023; published online December 15, 2023. Assoc. Editor: Gaurav Ameta.

point of failure can rapidly propagate throughout the system and result in quality issues with the final product, emphasizing the importance of robust cyberattack detection and localization measures. For instance, in a car assembly process, an attack on the machining stage could alter the dimensions of the part being assembled, leading to the final product failing to serve its intended purpose and introducing quality issues. Therefore, it is crucial to detect cyberattacks at an early stage and accurately localize the stage under attack.

Most existing studies on the cybersecurity of multistage manufacturing systems primarily focus on additive manufacturing (AM) contexts [10–13]. However, these methods are specific to AM and cannot be readily applied to generic MMS, such as assembly processes. For instance, in Ref. [10], each layer's printing is treated as a stage, and an alteration detection method based on image analysis after each layer is printed is developed. While such methods are applicable to layer-by-layer manufacturing processes, there is a lack of fundamental research that analyzes the propagation mechanism of cyberattacks to other stages and how this propagation can be leveraged for attack localization. Hence, our goal is to analyze attack propagation in a generic multistage process and develop a cyberattack detection algorithm that can be applied across multiple MMS applications.

This paper considers a generalized multistage manufacturing system, where each stage sequentially processes the product. Each stage consists of a set of sensors, which take measurements of the product, and a controller that calculates the control output based on the sensor measurements from the previous stage to ensure the output sensor measurements of the stage are at the desired level. Because of this control mechanism, the impact of false data injection (FDI) attacks in a specific stage can propagate to later stages. In other words, a false data injection in the previous stage may not cause a physical impact on the attacked stage but on later stages. Moreover, when the attacker has some knowledge about the system, the attack can be designed to be undetectable at that stage [14]. The above factors pose significant challenges in detecting and localizing the attack in MMS, which we aim to address in this paper. The contributions of the paper can be summarized as follows:

(1) We characterize a generic multistage manufacturing process model using Kalman filter (KF) and stochastic state-space model and perform theoretical analysis to extract the features uniquely related to the location of the attack.
(2) We design a hybrid detection framework that integrates the benefits of signature- and anomaly-based detection techniques that fulfill the localization function without relying on attack data for training. This is achieved based on the theoretical analysis, which extracts the feature based on the domain knowledge of the system dynamics.
(3) We designed a group regularization-based framework for simultaneous attack detection and localization. Unlike most existing methods that detect and localize in a two-phase manner, the group lasso-based framework enables us to identify the occurrence and location of potential false data injection attacks in real-time. With such information, further investigation and treatments can be triggered to minimize the attack's impact on the system and its users.

2 Literature Review

Manufacturing systems are commonly considered as examples of cyber-physical systems (CPSs) in the context of cybersecurity. Security approaches that are typically applicable to CPSs, such as vulnerability analysis [15–17], secure IoT network architecture design [18,19], intrusion detection [20,21], and cyberattack-resilient state estimation and control [22–24], also apply to manufacturing systems. In this section, we specifically focus on reviewing the literature on cyberattack detection in manufacturing systems, which can be categorized into model-based and data-driven techniques.

Model-based methods rely on the physics rules that govern manufacturing processes to identify anomalies in events that do not adhere to these rules. The emergence of digital twins has enabled the development of many model-based methods for cyberattack detection, thanks to their real-time, high-fidelity representation of the actual system.

The literature on data-driven cyberattack detection in manufacturing systems can be divided into two categories: (i) signature-based methods and (ii) anomaly-based methods. Signature-based methods operate by matching patterns with known attacks and triggering an alarm when a match is found [3,25–27,40]. In essence, they function similarly to supervised machine learning algorithms, where they are trained using normal and under-attack data, and then deployed on real-world data for attack detection [28–33]. For example, Song et al. propose a real-time attack detection system using a convolutional neural network in cyber-manufacturing systems for detecting defects [34]. Wu et al. utilized machine learning algorithms to detect cyber-physical attacks in cyber-manufacturing systems, simulating attacks on 3D printing and CNC machining [35]. Another work by Wu et al. focused on detecting malicious infill defects in 3D printing using image classification, where features are extracted from images and classification algorithms such as the naive Bayes classifier and J48 Decision Tree are applied [31].

Anomaly detection methods aim to identify patterns representing the expected behavior of the system and raise alarms upon recognizing significant deviations from the normal pattern. These methods function similarly to unsupervised learning algorithms [29,36–38]. For instance, Qian et al. proposed a scheme for detecting cyber attacks in the cyber and physical stages of SCADA systems using a nonparallel hyperplane-based fuzzy classifier [37]. Kwon et al. introduced a hybrid approach that combines anomaly and signature detection algorithms for detecting cyber attacks in physical systems. They use normal data in training datasets to establish a threshold and then apply the trained model to test datasets to evaluate its performance [29].

The proposed method in this paper combines signature and anomaly detection techniques. The signatures associated with attacks on each stage in the multistage system capture the correlation between sensor measurements from all stages and the injected data. However, unlike the supervised learning-based algorithms mentioned earlier, these signatures are not extracted purely from data-driven approaches. Instead, they are derived from the system dynamics based on domain knowledge. Therefore, the proposed method can also be considered as an anomaly-based detection algorithm that relies solely on normal data rather than attack data. This characteristic makes the proposed method more realistic and applicable in MMS applications where there is typically a lack of sufficient attack data.

3 System Representation

As discussed in Sec. 1, MMSs consist of multiple stages in which the output of stage i is the input to stage $i+1$. The representation of the system can be seen in Fig. 1. At each stage, the control action is taken based on the measurements of the input state to control the states to the reference values (setpoints). In practice, measurements before and after a processing stage maybe taken at the same station. However, we generalize the measurements to be taken after each stage, which means the sensor measurements is obtained from the output state at each stage. The attack we are considering in this paper is false data injection attack, where fake sensor measurements are sent to the controllers.

3.1 Multistage State-Space Model.

The state-space representation has been widely used in the literature to characterize multistage processes [39–42]. In this paper, we use a stochastic state-space model to represent an K-stage MMS, where the state transition and measurement functions vary between stages.

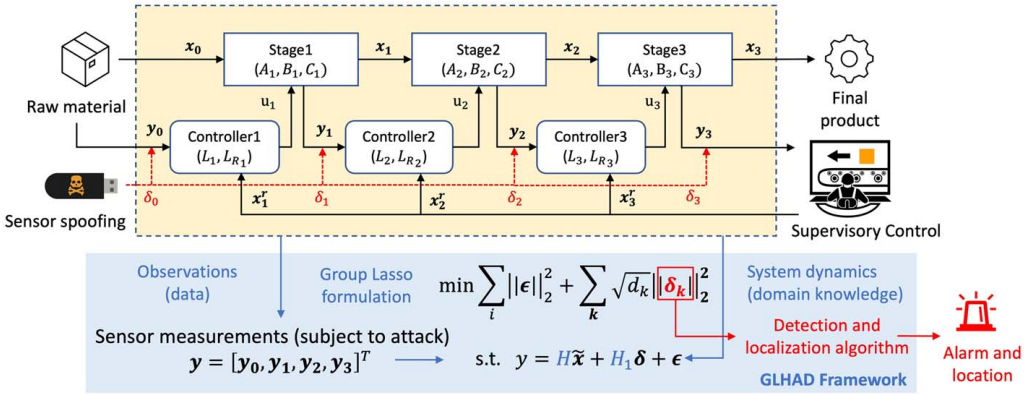

Fig. 1 The multistage manufacturing system and the proposed GLHAD framework

Let $\mathbf{x}_k$ denotes the state variable of the product at stage k such that $\mathbf{x}_k \in \mathbb{R}^{m_k}$, where m_k is the number of state variables at stage k. Let $\mathbf{y}_k$ denote the sensor measurements at stage k such that $\mathbf{y}_k \in \mathbb{R}^{n_k}$, where n_k is the number of sensors at stage k. In the case that the original sensor measurement is in the form of a profile or a video stream, the extracted features can be used to formulate the state-space model, which can be represented by multiple elements in the vector $\mathbf{y}_k$, and n_k is the total number of features, which can be viewed as virtual sensors at stage k. For continuous manufacturing processes, sensor data snapshots or sliding windows at a constant frequency can be used to extract discrete-time features to formulate the model. Let $\mathbf{u}_k$ denotes the control actions at time k such that $\mathbf{u}_k \in \mathbb{R}^{p_k}$, where p_k is the dimension of control action at stage k. $\mathbf{w}_k$ and $\mathbf{v}_k$ are the process noise and measurement noise at stage k, respectively, which are independent of all the other variables and are assumed to be not affected by any system anomaly, including attacks. Both $\mathbf{w}_k$ and $\mathbf{v}_k$ follow multivariate normal distributions with zero mean, i.e., $\mathbf{w}_k \sim N(0, W_k)$ and $\mathbf{v}_k \sim N(0, V_k)$, where $W_k \in \mathbb{R}^{m_k \times m_k}$ and $V_k \in \mathbb{R}^{n_k \times n_k}$ are the covariance matrices of the noise terms at stage k. The state-transition function and the measurement function are

$$\mathbf{x}_k = A_k \mathbf{x}_{k-1} + B_k \mathbf{u}_k + \mathbf{w}_k \tag{1}$$

$$\mathbf{y}_k = C_k \mathbf{x}_k + \mathbf{v}_k \tag{2}$$

In the above equations, $A_k \in \mathbb{R}^{m_k \times m_{k-1}}$, $B_k \in \mathbb{R}^{m_k \times p_k}$, and $C_k \in \mathbb{R}^{n_k \times m_k}$ are the system matrix, input matrix, and output matrix for stage k, respectively. For stage $k = 1$, the input state $\mathbf{x}_0$ represents the initial state of the product. For example, for a multistage machining process, $\mathbf{x}_0$ can be the dimensions of the part before processing. We assume the matrices A_k, B_k, C_k, V_k, W_k for each stage k are known. In Eq. (1), $A_k \mathbf{x}_{k-1}$ is the action taken on the input product (state variable) at stage k. Also, $B_k \mathbf{u}_k$ is the action taken at stage k by the controller to make sure that the state variable is the desired (reference) state variable at stage k.

Coupled with the state-space model, the Kalman filter is the optimal state estimator for the stochastic linear state-space model [43]. Therefore, we use a Kalman filter for state estimation. The Kalman gain (K_k) is derived based on the KF formulations of discrete time, and the Kalman filter formulas are given in the following:

$$\hat{\mathbf{x}}_{k|k-1} = A_k \hat{\mathbf{x}}_{k-1|k-1} + B_k \mathbf{u}_k \tag{3}$$

$$\hat{\mathbf{x}}_{k|k} = \hat{\mathbf{x}}_{k|k-1} + K_k \hat{\mathbf{y}}_k \tag{4}$$

$$\hat{\mathbf{y}}_k = \mathbf{y}_k - C_k \hat{\mathbf{x}}_{k|k-1} \tag{5}$$

In the above equations, $\hat{\mathbf{x}}_{k|k-1}$ denotes the predicted system state given at time $k-1$, $\hat{\mathbf{x}}_{k|k}$ denotes the updated state estimation given the measurement at time k, $\mathbf{y}_k$, and $\hat{\mathbf{y}}_k$ denotes the residual at time k, which is the difference between predicted and actual measurements, as shown in Eq. (5).

3.2 Controller Model. We consider a linear controller, where the control action at stage k is calculated as a linear function of the state estimate and the control parameters as follows:

$$\mathbf{u}_k = L_k \hat{\mathbf{x}}_{k-1|k-1} + L_{R_k} \mathbf{x}_k^r \tag{6}$$

In the equation above, $\hat{\mathbf{x}}_{k-1|k-1}$ is the state estimate of the product from the previous stage, $k-1$. (the type of state estimator can vary, but the state estimate is calculated based on the instant or historical sensor measurements. The commonly used Kalman filter state estimator is introduced below) $\mathbf{x}_k^r$ is the control setting parameters for controller k (in this paper, we assume $\mathbf{x}_k^r$ has the same dimensionality as $\mathbf{x}_k$). The linear controller calculates the control action as a linear combination of the estimated state and the control parameter. The matrices L_k and L_{R_k} are the linear coefficients of the estimated state after the previous stage and the reference values, respectively.

As a typical example, a linear quadratic Gaussian (LQG) controller follows the above formulation and can be used to calculate control action based on the state estimation ($\hat{\mathbf{x}}_{k|k}$) and the reference state of each stage ($\mathbf{x}_k^r$). The controller is calculated based on the minimization of

$$J = \mathbb{E}[\Sigma_{k=0}^{K} (\mathbf{x}_k - \mathbf{x}_k^r)^T U (\mathbf{x}_k - \mathbf{x}_k^r) + \Sigma_{k=0}^{K-1} \mathbf{u}_k^T Z \mathbf{u}_k]$$

where Z and U are positive-semi definite matrices defining the cost. Under the LQG setting, Eq. (6) is defined by

$$L_k = (B_k^T S_{k+1} B_k + Z)^{-1} B_k S_{k+1} A_k$$

where S_k is calculated by the following matrix Riccati difference equation that runs backward in time: $S_k = A_k^T (S_{k+1} - S_{k+1} B_k (B_k^T S_{k+1} B_k + Z)^{-1} B_k S_{k+1}) A_k + U$, $S_K = F$.

3.3 False Data Injection. In this paper, we consider the generic FDI attacks on the sensor data. FDI attacks are a type of data integrity attack that tampers with (modify or replace) the sensor output in order to disrupt the system operations. To implement an FDI attack, the attacker usually exploits the target system's OT vulnerabilities, gains unauthorized access, manipulates the data, and employs techniques to avoid detection. The injected false data can cause various effects depending on the attacker's goals, such as causing incorrect control decisions, triggering alarms, or masking other malicious activities [44–46]. In this paper, we assume that FDI attacks occur at the stage level, meaning they target one or several stages rather than the entire MMS. This assumption is practical because most MMS consist of

diverse equipment that relies on various network protocols, necessitating extensive knowledge of the entire MMS for an attacker to execute a successful FDI attack. Moreover, in many large-scale settings, attacking multiple stages becomes infeasible due to the stages being geographically dispersed. In this context, a false data injection attack on stage k can be modeled as

$$\mathbf{y}_k^a = \mathbf{y}_k + \delta_k \tag{7}$$

where δ_k represents the bias introduced to the sensor measurements by FDI at stage k. $\mathbf{y}_k^a$ is the vector of the under-attack sensor measurements at stage k. Notice that δ_k is introduced to facilitate theoretical analysis. The attacker may implement the FDI attack by replacing the original sensor measurement with $\mathbf{y}_k^a$. Hence, δ_k does not represent the data being injected, but only the bias caused by the false data injection.

4 Methodology

Given the system model described in Sec. 3, we propose a group Lasso-based hybrid attack detection and localization (GLHAD) framework against false data injection attacks in an MMS. The framework is depicted in Fig. 1. It takes the sensor data and the system dynamics of the MMS as input and integrates them into a group Lasso formulation. The GLHAD framework is based on theoretical analysis of the FDI attack propagation and characterization of the attack propagation as data features extracted from the system analysis. The major advantage of GLHAD is that it provides a generalizable framework to not only detect the occurrence of the attack, but also localize the source of the attack. To achieve this, we first perform theoretical analysis in Sec. 4.1 to build the mathematical model, based on which we extract the data features characterizing the impact of false data injection attacks on the sensor data. We then use the characterization to extract signatures from state estimation residuals that indicate the occurrence of the attack in the system and location in terms of the under-attack stage in the system. The formulation of the group Lasso model and the proposed attack detection and localization algorithm is described in Sec. 4.2.

4.1 Theoretical Analysis.

To facilitate our analysis, we introduce the *augmented state variable* $\tilde{\mathbf{x}}$ comprised of the input state $\mathbf{x}_0$, representing the state of the input material, and the reference values, $\mathbf{x}_k^r$ at each stage k.

$$\tilde{\mathbf{x}} = [\mathbf{x}_0^T \ \cdots \ (\mathbf{x}_K^r)^T]^T \tag{8}$$

where $\mathbf{x}_0$ represents the input of the system, also x_k^r represents the reference state variable at stage k. Notice the state variables $\mathbf{x} = [\mathbf{x}_0^T, \mathbf{x}_1^T \cdots \mathbf{x}_N^T]^T$ tracks the product state at each stage, and with the control actions at each stage calculated based on the measurements after the previous stage, the $\mathbf{x}_i$'s are cross-correlated. On the other hand, the augmented state variables are the independent external inputs to the MMS at each stage under our system setting. In this context, the sensor measurements, $\mathbf{y} = [\mathbf{y}_0{}^T, \mathbf{y}_1^T \ \cdots \ \mathbf{y}_K{}^T]^T$, with $\mathbf{y}_k$ representing the sensor measurements at stage k, for $k \in \{0, \ldots, K\}$, can be represented as a linear function of $\tilde{\mathbf{x}}$ as follows:

$$\mathbf{y} = H\tilde{\mathbf{x}} + \varepsilon \tag{9}$$

where H is the augmented measurement matrix. Specifically, H characterizes the relationship between the augmented state variables and the sensor outputs. H is comprised of $(K+1)\times(K+1)$ submatrices h_{ij}, characterizing the relationship between the sensor measurements in stage $i-1$ and the augmented state in stage $j-1$. We develop Proposition 1 to define matrix H.

Proposition 1. *For an MMS described in Sec. 3, the matrix H in Eq. (9) can be written as*

$$H = [h_{ij}]_{(K+1)\times(K+1)} \tag{10}$$

For $j \geq 2$, we have

$$h_{ij} = \begin{cases} 0, & i = 1, \ldots, j-1 \\ C_{i-1}B_{i-1}L_{R_{i-1}}, & i = j \\ C_i \prod_{b=j}^{i} (A_b + B_b L_b) B_1 L_{R_1}, & i = j+1, \ldots, K \end{cases}$$

For $j = 1$, we have

$$h_{i1} = \begin{cases} C_0, & i = 1 \\ C_1[A_1 + B_1 L_1 K_0 C_0], & i = 2 \end{cases}$$

For $i \geq 3$, we have

$$\begin{aligned} h_{i1} = C_{i-1} \Bigg[& \prod_{k=1}^{i-2} (A_{i-k} + B_{i-k}L_{i-k})(A_1 + B_1 L_1 K_0 C_0) \\ & - \cdots \sum_{j=1}^{i-3} \Bigg[\prod_{k=1}^{j} (A_{i-k} + B_{i-k}L_{i-k})(B_{i-(j+1)}L_{i-(j+1)}) \\ & \cdots \prod_{c=j+2}^{i-1} (I - K_{i-c}C_{i-c})A_{i-c}(I - K_0 C_0) \Bigg] \\ & - \cdots (B_{i-1}L_{i-1}) \prod_{m=2}^{i-1} (I - K_{i-m}C_{i-m})A_{i-m}(I - K_0 C_0) \Bigg] \end{aligned}$$

The proof is provided in Appendix A.

Remark 1. Proposition 1 defines sensor measurements $\mathbf{y}$ as a function of the augmented state variable $\tilde{\mathbf{x}}$, representing the initial product state and the reference values of the distributed stage-level control in an MMS. This new system model distinguishes the deterministic external inputs, assumed to be immune to false data injection, from the intermediate product states between stages that are affected by the false data injection. Using this representation, we can estimate the augmented state variables based on sensor data and obtain the state estimation residuals, which helps us in detecting and localizing the attack. ■

When the system is under attack, based on the linearity assumptions in the system dynamics and controller model, the attack on stage k can propagate to stage $k+1$ through the control mechanism in Eq. (6). This can be captured by the sensor data in stage $k+1$, which then further affects the control in later stages. To characterize this linear propagation, we construct a new model for system under attack, characterized by the Proposition below.

Proposition 2. *For an MMS described in Sec. 3, under a false data injection attack characterized by vector* $\delta = [\delta_0{}^T, \delta_1^T \cdots \delta_K^T]^T$, *where* δ_k *represents the false data injected at stage k, the sensor measurement* $\mathbf{y}$ *can be expressed as*

$$\mathbf{y} = H\tilde{\mathbf{x}} + H_1\delta + \varepsilon \tag{11}$$

$$H_1 = [\tilde{h}_{ij}]_{(K+1)\times(K+1)} \tag{12}$$

where

$$\tilde{h}_{ij} = \begin{cases} 0, & i = 1, \ldots, j-1 \\ I, & i = j \\ C_{i-1}B_{i-1}L_{i-1}K_{j-1}, & i = j+1 \end{cases} \tag{13}$$

for $i \geq j+2$

$$\tilde{h}_{ij} = C_{i-1}\Bigg[\prod_{m=1}^{i-(j+1)}(A_{i-m}+B_{i-m}L_{i-m})B_jL_j + \cdots \sum_{c=1}^{i-(j+2)}\Bigg[\prod_{m=1}^{c}(A_{i-m}+B_{i-m}L_{i-m})B_{i-(c+1)}L_{i-(c+1)} \cdots \prod_{b=c+2}^{i-j}(I-K_{i-b}C_{i-b})A_{i-b}\Bigg] + \cdots B_{i-1}L_{i-1}\prod_{c=2}^{i-j}(I-K_{i-c}C_{i-c})A_{i-c}\Bigg]K_j$$

The proof is provided in Appendix B.

Remark 2. In Proposition 2, H_1 characterizes the relationship between the sensor measurements and the injected false data, with consideration of the attack propagation resulted from the multistage process. Equation (12) will facilitate extracting the important features from $\mathbf{y}$ to accurately detect FDI attacks and localize the source stage of the attack. Based on the linear relationship, we will develop the GLHAD framework based on the features extracted from the state estimation residuals. ■

To detect anomalous patterns in state estimation residuals, we must analyze the variance of $\mathbf{y}$. In Eq. (9), since $\tilde{\mathbf{x}}$ contains only input variable $\mathbf{x}_0$, and all reference values are deterministic, process noise, and measurement noise $\mathbf{w}_k$ and $\mathbf{v}_k$ contribute to the noise term ε. Thus, we derive the variance of ε in Proposition 3, which will be used later to analyze ε and identify abnormal patterns caused by false data injection.

PROPOSITION 3. *For an MMS described in Sec. 3, denote the block diagonal matrices of process and measurement noise covariance as $\Sigma_x = diag(W_1, \ldots, W_K)$ and $\Sigma_y = diag(V_1, \ldots, V_K)$, respectively. The covariance of ε in Eq. (9) follows:*

$$\Sigma_\varepsilon = H_w\Sigma_x H_w^T + H_1\Sigma_y H_1^T \tag{14}$$

where H_w can be represented as: $H_w = [h''_{ij}]_{(K+1)\times(K+1)}$. In the above expression, for any j, we have

$$h''_{ij} = \begin{cases} 0, & i = 1, \ldots, j-1 \\ C_{i-1}, & i = j \\ C_{i-1}[A_{i-1} + B_{i-1}L_{i-1}K_{i-2}C_{i-2}], & i = j+1 \end{cases}$$

for $i \geq j+2$

$$h''_{ij} = C_{i-1}\Bigg[\prod_{m=1}^{i-(j+1)}(A_{i-m}+B_{i-m}L_{i-m})(A_j+B_jL_jK_{j-1}C_{j-1}) \cdots \sum_{c=1}^{i-(j+2)}\Bigg[\prod_{m=1}^{c}(A_{i-m}+B_{i-m}L_{i-m})B_{i-(c+1)}L_{i-(c+1)} \cdots \prod_{b=c+2}^{i-j}(I-K_{i-b}C_{i-b})A_{i-b}(I-K_{j-1}C_{j-1})\Bigg] + \cdots B_{i-1}L_{i-1}\prod_{c=2}^{i-j}(I-K_{i-c}C_{i-c})A_{i-c}(I-K_{j-1}C_{j-1})\Bigg]$$

Remark 3. Proposition 3 derives the expression of Σ_ε, which will be used for formulating hypothesis testing on ε. By understanding the normal covariance matrix, the anomalous pattern in the data caused by the attack will lead to a rejection of null hypothesis that can be used to identify the attack. The stage-level hypothesis tests formulated based on Σ_ε will be used to localize the attack. ■

4.2 The GLHAD Framework. Based on the theoretical analysis in Sec. 4.1, we propose a GLHAD framework that fuses the sensor data and domain knowledge of the system dynamics into the group Lasso formulation. The GLHAD framework combines the advantage of both signature-based method and anomaly detection methods, where we can identify the location of the attack using a signature-based mechanism without relying on a comprehensive dataset that contains labeled attack data to learn the features of attacks at different locations.

The GLHAD framework consists of two steps. The first step is the group lasso formulation, and the second step is the online detection and localization. In the first step, we use the augmented state estimation to distinguish the variance caused by external input and the variance caused by attack propagation to extract the state estimation residuals that serves as the input to the group lasso formulation. In the second step, we solve the group lasso problem and run the sequential hypothesis test based on the solution and raise alarms of attacks and with the attack location information.

According to Proposition 1, under normal condition, the system model follows Eq. (9). The estimated augmented state variable, $\hat{\mathbf{x}}$, is calculated by projecting $\mathbf{y}$ onto the column space spanned by H

$$\hat{\mathbf{x}} = (H^TH)^{-1}H^T\mathbf{y}$$

and the state estimation residual, $\mathbf{r}$, is calculated as

$$\mathbf{r} = \mathbf{y} - \hat{\mathbf{y}}$$

where $\mathbf{r} = [\mathbf{r}_0^T \ \cdots \ \mathbf{r}_K{}^T]^T$, and $\mathbf{r}_k$ is the state estimation residual at stage k. Under normal condition (when there is no attack), the augmented state estimate should be close to the true augmented state, and the residuals should be close to zero. On the other hand, based on Proposition 2, the residuals will have a component spanned by the matrix H_1, whose expectation is not 0. Therefore, the magnitude of the residual $\mathbf{r}$ is an indication of whether the system is under attack. To further infer the location of attack, we need to analyze the patterns in $\mathbf{r}$ that are consistent with attack propagation mechanism. We acknowledge that the $\mathbf{r}$ will not fully reconstruct the term $H_1\delta + \varepsilon$, as it is obtained by projecting y onto the column space of H, but rather the subspace of H_1 that is orthogonal to the column space of H. Intuitively speaking, $\mathbf{r}$ represents the part in $\mathbf{y}$ that is "unexplainable" by H. Therefore, it is necessary to identify the basis of such subspace of H_1 that is orthogonal to H as well as separating the subspaces corresponding to attacks on different stages. Denote R_k as the subspace basis for attack on stage k, it can be obtained by projecting the columns of H_1 corresponding to stage k on to the columns space of H and taking the residue

$$R_k = H_1^k - H(H^TH)^{-1}H^TH_1^k \tag{15}$$

Note that R_k may not be full rank. Therefore, we apply principal component analysis to obtain the principal components of R_k, which formulates the signature basis of stage k, $\tilde{R}_k$. In other words, $\tilde{R}_k = R_k[\mathbf{e}_1, \mathbf{e}_2, \ldots, \mathbf{e}_{g_k}]$, where $\mathbf{e}_i$ is the ith eigenvector of $(R_k)^TR_k$, and the number of principal components g_k ($g_k \leq m_k - n_k$) is selected based on the eigenvectors. After obtaining the signature basis $\tilde{R} = [\tilde{R}_1, \ldots, \tilde{R}_K]$, Eq. (12) can be rewritten as

$$\mathbf{y} = H\tilde{\mathbf{x}} + \sum_{k=1}^{K}H_1^k\delta_k + \varepsilon \tag{16}$$

and hence, residual $\mathbf{r}$ can be written as

$$\mathbf{r} = \sum_{k=1}^{K}\tilde{R}_k\tilde{\delta}_k + \tilde{\varepsilon} \tag{17}$$

In the above equations, the transformed FDI vector $\tilde{\delta}^k$ is a linear projection of δ such that $\tilde{R}_k\tilde{\delta}_k \approx H_1^k\delta_k$, the signature basis $\tilde{R}_k$ is orthogonal to H, and $\tilde{\varepsilon}$ is the new noise term that is "unexplainable" by either H or $\tilde{R}$. Based on the above model, the FDI vector

$\delta = [\delta_1, \ldots, \delta_K]$ can be calculated by projecting the residuals $\mathbf{r}$ onto the column space of R. Based on our assumption that there is only one stage under attack, the vector δ should be group-sparse, where each group is a stage and corresponds to a δ_i, and only one of the δ_i is non-zero. As each $\tilde{\delta}^k$ is a linear transformation of δ_k, the vector $\tilde{\delta}$ should preserve the sparsity of δ. Therefore, the detection and localization problem can be formulated as a group Lasso problem, where we estimate the transformed FDI vector $\tilde{\delta}$ by regressing the state estimation residual $\mathbf{r}$ against the signature basis $\tilde{R}_k$. The group sparsity of the obtained coefficients will indicate the location of the attack, and the magnitude of the residuals spanned by the signature basis of the group can be used to detect the attack.

The group Lasso formulation can then be written as follows:

$$\min_{\hat{\delta}_k \in \mathbb{R}^{g_k}} \left\| \mathbf{r} - \sum_{k=1}^{K} \tilde{R}_k \hat{\delta}_k \right\|_2^2 + \lambda \sum_{k=1}^{K} \sqrt{g_k} \left\| \hat{\delta}_k \right\|_2^2 \tag{18}$$

$$\text{s.t.} \quad \mathbf{y} = H\hat{\mathbf{x}} + \mathbf{r} \tag{19}$$

The above groups Lasso formulation minimizes the error term $\tilde{\varepsilon}$ while enforcing the group sparsity in $\hat{\delta}$ by penalizing the group l_2 norm. λ is a positive tuning parameter that decides how much sparsity is enforced, where a smaller λ will result in a more sensitive detection algorithm which can potentially generate more false alarms, while a larger λ leads to a more robust detection algorithm that may miss some of the slight attacks. Our λ is tuned based on the training data containing only the normal data, to achieve a desired type-I error rate. In practice, λ can also be selected based on the smallest magnitude of δ to be detected.

After solving the above group Lasso problem, the estimated residual is calculated as follows:

$$\hat{\mathbf{r}}_k = \tilde{R}_k \hat{\delta}_k \tag{20}$$

We then test the magnitude of the residuals using a χ^2 test based on the covariance matrix of the noise term ε given in Proposition 3. As the signatures were processed by principal component analysis, the covariance matrix used in the stage-level χ^2 test is calculated as

$$\tilde{\Sigma}_k = \Psi_k^T \Phi^T \Sigma_\varepsilon \Phi \Psi_k^T \tag{21}$$

where $\Phi = I - H(H^T H)^{-1} H^T$ and $\Psi_k = I - \tilde{R}_k \tilde{R}_k^T$. The stage-level χ^2 statistic is calculated as

$$\chi_k^2 = \hat{\mathbf{r}}_k^T \tilde{\Sigma}_k^{-1} \hat{\mathbf{r}}_k \tag{22}$$

The χ_k^2 is compared with the upper α quantile of the χ^2 distribution with g_k degrees-of-freedom, $\chi_{\alpha,k}^2$. An alarm is triggered when $\chi_k^2 > \chi_{\alpha,k}^2$. This mechanism also potentially allows us to detect attacks on multiple stages simultaneously, but we do not consider such case both because of practicality and the complexity of the analysis involving multiple correlated hypothesis tests. In this paper, under the single stage-under-attack (SUA) assumption, the stage with the lowest p-value is considered as the under-attack stage when multiple alarms are triggered. The pseudo-code of the detection algorithm is given in Algorithm 1.

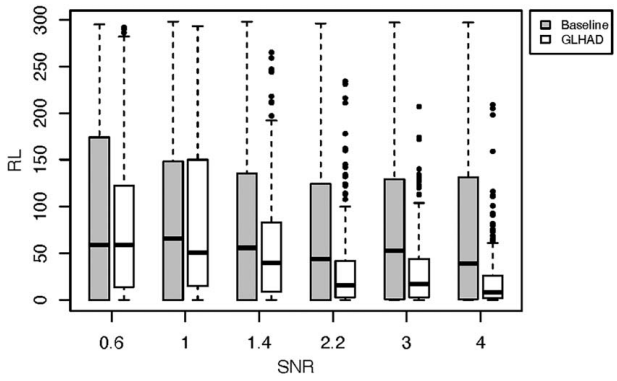

Fig. 2 Box-plot comparison of detection delays (run lengths) for the baseline method (group-wise χ^2 test) and GLHAD in the numerical study; shaded boxes represent the baseline method, while blank boxes indicate the GLHAD framework

Algorithm 1: GL-based attack detection and identification for multistage linear system

Input: $\mathbf{y}$, H, $\tilde{R}_k$, $\tilde{\Sigma}_k$ $(k = 1, \ldots, K)$, λ, α

$\hat{\mathbf{x}} \leftarrow (H^T H)^{-1} H^T \mathbf{y}$; $\mathbf{r} \leftarrow \mathbf{y} - H\hat{\mathbf{x}}$
Solve the group Lasso problem in Eq. (18)
$k \leftarrow 1$, $l \leftarrow 0$, $p_{min} \leftarrow 1$
while $k \leq K$ **do**
  $\hat{\mathbf{r}}_k \leftarrow \tilde{R}_k \hat{\delta}_k$
  $\chi_k^2 = \hat{\mathbf{r}}_k^T \tilde{\Sigma}_k^{-1} \hat{\mathbf{r}}_k$
  if $\chi_k^2 > \chi_{\alpha,k}^2$ **then**
    $p \leftarrow F_{\chi_{gk}^2}^{-1}(\chi_k^2)$
    if $p < p_{min}$ **then**
      $l \leftarrow k$
    end if
  end if
  $k \leftarrow k + 1$
end while
if $l = 0$ **then**
  return no attack
else
  return stage-l is under attack
end if

Algorithm 1 shows that, after solving the group Lasso problem, the detection and localization process are completed simultaneously by running the stage-level χ^2 tests. The algorithm demonstrates a combination of the theoretical analysis which involves the derivation of the signature basis and statistical learning which involves the group lasso algorithm based on the sensor data. The performance of the above algorithm is tested based on a numerical study on randomly generated system parameters as well as a case study based on an assembly process.

5 Experimental Results

In this section, we conduct a numerical study and a case study to compare the performance of the proposed GLHAD framework with the baseline method—a stage-level χ^2 test based on the group-wise hypothesis testing scheme. In the baseline method, the residuals are obtained based on the measurement function (2)

$$\gamma_k = (I - C_k(C_k^T C_k)^{-1} C_k^T)\mathbf{y}_k \tag{23}$$

The covariance matrix of γ_k can be calculated as

$$\Sigma_k = (I - (C_k^T C_k)^{-1} C_k^T) V_k (I - (C_k^T C_k)^{-1} C_k^T)^T$$

Notice that Σ_k is based on the independent measurement function of each stage without involving the stage-stage interaction. Then, we apply the stage-level χ^2 test on each stage, where the test statistic is calculated as $\chi_{b,k}^2 = {\gamma_k}^T \Sigma_k^{-1} \gamma_k$. For stage k, an alarm is triggered when $\chi_{b,k}^2 > \chi_{\alpha,n_k}^2$. α is the type-I error and n_k is the dimension of $\mathbf{x}_k$.

5.1 Numerical Study. We consider an MMS consisting of three stations, where the state variable is three-dimensional, and each stage has five sensors. We randomly generate the elements in matrices (A_k, B_k, C_k with $n_k = 3$, $m_k = 5$ for $k = 0, 1, 2, 3$) from a standard normal distribution. The process noise and measurement noise at each stage come from the multivariate normal

distribution with mean zero and covariances: $W_k = 0.1I$, $V_k = 0.1I$ for $k = 0, \ldots, 3$. Three hundred replications were implemented under the normal scenario for us to tune the parameter. Then, we simulate 100 replications for each of the sensor attacks on each stage with different attack severities and compare the performance of the two methods in terms of detection delay and localization accuracy. The severity of attack is quantified by the signal-to-noise ratio (SNR) calculated as follows:

$$\text{SNR} = \sqrt{\delta_k^T V_k^{-1} \delta_k}$$

Six different levels of SNR are considered in the numerical study: SNR $\in \{0.1, 1, 1.4, 2.2, 3, 4\}$.

The expected detection delay is shown in terms of the run length (RL) indicating the number of observations taken to raise the first alarm since the onset of the attack. The RLs of 100 replications are shown with box plots in Fig. 2. The type-I error rate is chosen at 0.005, so the nominal average run length (ARL) is set to be 200. It is shown that both methods are sensitive to attacks, and the detection delay decreases as the attack severity increases. However, the proposed GLHAD framework shows a lower ARL and a smaller variance in the run length distribution, which indicates a more timely and reliable attack detection.

Figure 3 shows the overall localization accuracy, which is calculated as the probability of correctly identifying the true SUA. The shaded area indicates the 90% confidence interval if the attack accuracy. The results show that the proposed GLHAD outperforms the baseline method, especially when the attack occurs at an earlier stage. This can be explained by the fact that the GLHAD framework takes the attack propagation into consideration and is hence utilizing more information from later stages than the independent stage-level hypothesis tests in the baseline method. It is also not surprising as the attack severity increases, the localization accuracy also increases. Tables 1 and 2 show the confusion matrix of the localization results. A key observation is that the baseline method misclassifies attacks at early stages (e.g., stage 0) as attacks at later stages (first column in Table 2). In contrast, the GLHAD framework has a lower probability of misclassifying an attack as attacks in later stages, but has a slightly higher chance of misclassifying them as early-stage attacks.

5.2 Case Study. In the case study, we implement the GLHAD framework and benchmark method on a real-world three-stage SUV side frame assembly process introduced by Ref. [47]. The assembly process creates the final product called "inner-panel-complete." This product consists of four main components: the A-pillar, B-pillar, rail roof side panel, and rear quarter panel. These components are assembled across three stations, namely Stations I, II, and III. The state variable $\mathbf{x}_k$ represents the accumulated part deviation up to station k, the input $\mathbf{u}_k$ is the fixture deviation where we assume to be controlled by an LQG controller, and $\mathbf{y}_k$ is the measurement deviation observed at station k. The configuration of the system is described through matrices A, B, and C, all of which are determined by the process and product design. Matrix A, referred to as the dynamic matrix, outlines how the orientation of the assembly changes as the individual parts are transferred between the different stations. Essentially, A captures the alterations in part positioning that occur as the production process moves from one station to another. Matrix B serves as the input matrix, and it governs how deviations in the fixture (the tooling used for alignment) impact the overall deviation of the assembled part. This is contingent on the specific layout of the fixture and how it interacts with the part's geometry. More details can be found in Ref. [47].

We use the system parameters provided in Refs. [9,47] and consider only the first six state variables in our study. A 0.01 factor is used to re-scale the system to ensure computation efficiency considering the iterative substitutions in the matrix derivations, and the SNR values are selected to be 10, 15, 20, 30, 40, 50 in this study.

Table 1 Attack localization accuracy—GLHAD (synthetic data, SNR = 1.4)

		Actual SUA			
		S0	S1	S2	S3
Predicted SUA	S0	**0.766**	0.110	0.106	0.131
	S1	0.056	**0.839**	0.073	0.109
	S2	0.089	0.034	**0.756**	0.117
	S3	0.089	0.017	0.065	**0.642**

Note: Bold values are highest in each column.

Table 2 Attack localization accuracy—group-wise χ^2 test (synthetic data, SNR = 1.4)

		Actual SUA			
		S0	S1	S2	S3
Predicted SUA	S0	**0.387**	0.109	0.168	0.336
	S1	0.257	**0.313**	0.160	0.271
	S2	0.152	0.136	**0.432**	0.280
	S3	0.165	0.128	0.173	**0.534**

Note: Bold values are highest in each column.

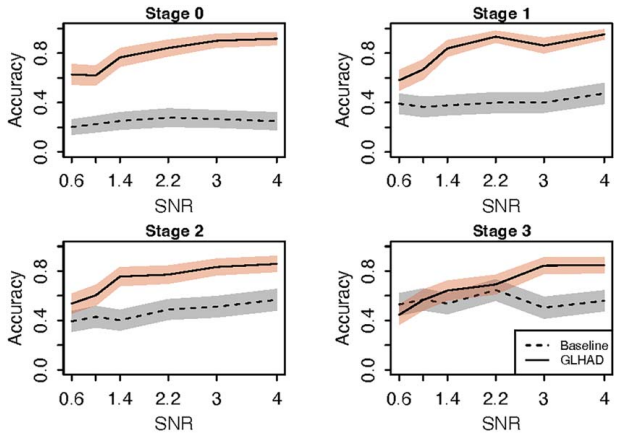

Fig. 3 Comparing localization accuracy for the baseline method (group-wise χ^2 test) and GLHAD when facing attacks of varying stages and magnitudes in the numerical study; the dashed lines denote the baseline method, and the solid lines indicate the GLHAD framework. The shaded areas represent the 90% confidence interval.

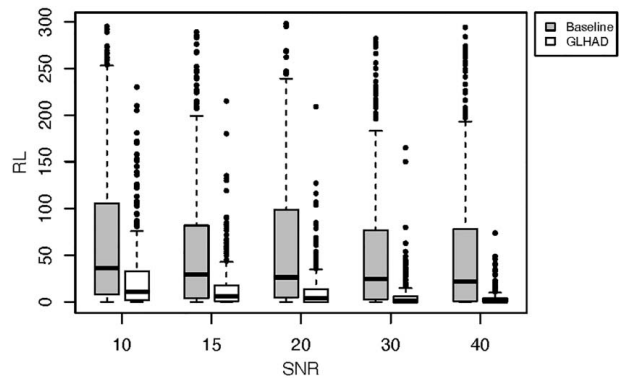

Fig. 4 Box-plot comparison of detection delays (run lengths) for the baseline method (group-wise χ^2 test) and GLHAD in the case study; gray boxes represent the baseline method, while white boxes indicate the GLHAD framework

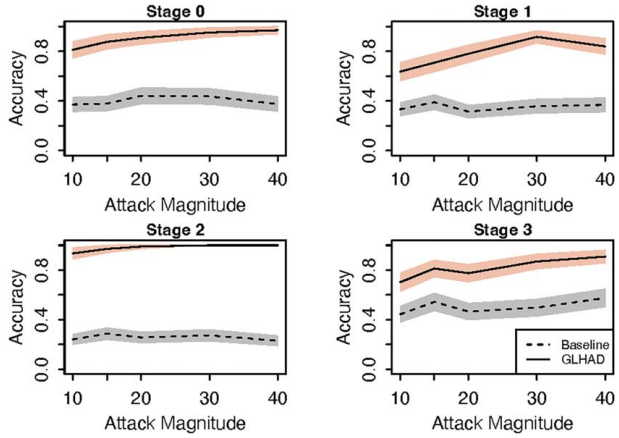

Fig. 5 Comparing localization accuracy for the baseline method (group-wise χ^2 test) and GLHAD when facing attacks of varying stages and magnitudes in the case study; the dashed lines denote the baseline method, and the solid lines indicate the GLHAD framework. The shaded areas represent the 90% confidence interval.

Table 3 Attack localization accuracy—GLHAD (case study, SNR = 2)

		Actual SUA			
		S0	S1	S2	S3
Predicted SUA	S0	**0.909**	0.086	0.000	0.093
	S1	0.045	**0.781**	0.010	0.085
	S2	0.027	0.070	**0.990**	0.047
	S3	0.018	0.063	0.000	**0.775**

Note: Bold values are highest in each column.

Table 4 Attack localization accuracy—group-wise χ^2 test (case study, SNR = 2)

		Actual SUA			
		S0	S1	S2	S3
Predicted SUA	S0	**0.442**	0.228	0.217	0.160
	S1	0.172	**0.315**	0.255	0.218
	S2	0.116	0.185	**0.257**	0.155
	S3	0.270	0.272	0.271	**0.466**

Note: Bold values are highest in each column.

The run length distributions, localization accuracy are shown in Figs. 4 and 5, respectively. The results demonstrate a more significant advantage of the proposed GLHAD framework than the baseline method both in detection delay and localization accuracy, especially when the attack is on stage 2. The baseline method fails to detect the attack with almost 80% probability, while the proposed GLHAD framework not only detects the attack, but also accurately localize the attack. This can also be observed in the confusion matrices in Tables 3 and 4. The GLHAD framework accurately identifies the SUA, while the baseline method has a lower than 50% localization accuracy, with similar mis-classification rates to regardless of stages.

6 Conclusion

In this paper, we proposed a novel group lasso-based hybrid cyberattack detection and localization framework for multistage manufacturing processes. Under a linear system dynamics assumption, we proposed a new system representation for the MMS by introducing an augmented state variable which helps distinguish the deterministic external control input to the MMS and the internal variations caused by the attack and attack propagation. We consider the most common sensor spoofing attack—false data injection in this paper. We propose the GLHAD framework based on a group lasso formulation that combines the advantages of signature-based and data-driven learning-based methods for attack detection. The formulation integrates the domain knowledge of the system dynamics into the sensor data-based group lasso algorithm, which allows us to fully utilize the real-time information from the data and the prior knowledge from the system. The group lasso formulation also allows us to detect the attack and localize it to the correct stage simultaneously with high accuracy. A numerical study and a case study validates the proposed framework and shows the advantage compared with traditional stage-level hypothesis testing methods. For future research, we aim to extend this work to consider nonlinear system representations, other types of attacks in the MMS, and relaxing the assumption of attack on a single stage.

Conflict of Interest

There are no conflicts of interest.

Data Availability Statement

The authors attest that all data for this study are included in the paper.

Appendix A: Proof of Proposition 1

We first consider the scenario when $j \in \{2, 3, \ldots, K\}$. We present a proof for $j = 2$. This proof can be easily generalized for any other $j \in \{3, \ldots, K\}$. h_{i2} represents the coefficient of $\mathbf{x}_1^r$ in $\mathbf{y}_{i-1}$. It can be easily seen that : $h_{12} = 0$, $h_{22} = C_1 B_1 L_{R_1}$. For $n \geq 2$, we use induction to complete the proof. For $i = 3$, h_{32} represents the coefficient of $\mathbf{x}_1^T$ in $\mathbf{y}_2$ and we have: $h_{32} = C_2[(A_2 + B_2 L_2]B_1 L_{R_1}$. We assume that for an arbitrary $i \geq 3$

$$h_{i2} = C_{i-1} \prod_{b=2}^{i-1} (A_b + B_b L_b) B_1 L_{R_1}$$

We must show that $h_{(i+1)2} = C_i \prod_{b=2}^{i} (A_b + B_b L_b) B_1 L_{R_1}$. From the assumption, it can be seen that: $\rho_{1,i-2} = \prod_{b=2}^{i-2} (A_b + B_b L_b) B_1 L_{R_1}$, where $\rho_{i,j}$ is the coefficient of x_i^r in $hat\mathbf{x}_{j\,|\,j}$. Hence, we can conclude that

$$\begin{aligned} \rho_{1,i-1} &= (I - K_{i-1} C_{i-1})(A_{i-1} + B_{i-1} L_{i-1}) j_2 + K_{i-1} h_{i2} \\ &= (A_{i-1} + B_{i-1} L_{i-1}) j_2 \\ &= \prod_{b=2}^{i-1} (A_b + B_b L_b) B_1 L_{R_1} \end{aligned}$$

We know that $\mathbf{y}_i = C_i \mathbf{x}_i = C_i A_i \mathbf{x}_{i-1} + C_i B_i \mathbf{u}_i = C_i A_i \mathbf{x}_{i-1} + C_i B_i [L_i \hat{\mathbf{x}}_{i-1|i-1} + L_{R_i} x_i^T]$. From assumption, it can be seen that $\beta_{1,i-1} = \prod_{b=2}^{i-1} (A_b + B_b L_b) B_1 L_{R_1}$. $\beta_{i,j}$ represents the coefficient of x_i^r in x_j. Hence, we can conclude that

$$h_{(i+1)2} = C_i A_i \beta_{1,i-1} + C_i B_i L_i \rho_{1,i-1} = C_i \left[\prod_{b=2}^{i} (A_b + B_b L_b) B_1 L_{R_1} \right]$$

When $j = 1$, h_{i1} represents the coefficient of $\mathbf{x}_0$ in $\mathbf{y}_{i-1}$. Hence, h_{11} and h_{12} represent the coefficient of $\mathbf{x}_0$ in y_0 and y_1, respectively. We use induction for the proof for $i \geq 3$.

h_{31} represents the coefficient of $\mathbf{x}_0$ in $\mathbf{y}_2$, so we have $h_{31} = C_2[(A_2 + B_2 L_2)(A_1 + B_1 L_1 K_0 C_0) - B_2 L_2 (I - K_1 C_1) A_1 (I - K_0 C_0)]$. We assume

that for an arbitrary $i \geq 3$, we have

$$h_{i1} = C_{i-1}\Bigg[\prod_{k=1}^{i-2}(A_{i-k}+B_{i-k}L_{i-k})(A_1+B_1L_1K_0C_0) - \cdots \sum_{j=1}^{i-3}\Bigg[\prod_{k=1}^{j}(A_{i-k}+B_{i-k}L_{i-k})(B_{i-(j+1)}L_{i-(j+1)}) \cdots \prod_{c=j+2}^{i-1}(I-K_{i-c}C_{i-c})A_{i-c}(I-K_0C_0)\Bigg] - (B_{i-1}L_{i-1}) \cdots \prod_{m=2}^{i-1}(I-K_{i-m}C_{i-m})A_{i-m}(I-K_0C_0)\Bigg]$$

From the assumption, it can be seen that

$$\theta_{i-2,i-1} = \prod_{k=2}^{i-2}(A_{i-k}+B_{i-k}L_{i-k})(A_1+B_1L_1K_0C_0) - \cdots \sum_{j=1}^{i-3}\Bigg[\prod_{k=2}^{j}(A_{i-k}+B_{i-k}L_{i-k})(B_{i-(j+1)}L_{i-(j+1)}) \cdots \prod_{c=j+2}^{i-1}(I-K_{i-c}C_{i-c})A_{i-c}(I-K_0C_0)\Bigg] - \cdots \prod_{m=2}^{i-1}(I-K_{i-m}C_{i-m})A_{i-m}(I-K_0C_0)$$

where $\theta_{i,j}$ is the coefficient of $\mathbf{x}_0$ of $\hat{\mathbf{x}}_{i|i}$ in y_j. Also, it can be seen that

$$\pi_{i-1} = \prod_{k=1}^{i-2}(A_{i-k}+B_{i-k}L_{i-k})(A_1+B_1L_1K_0C_0) - \cdots \sum_{j=1}^{i-3}\Bigg[\prod_{k=1}^{j}(A_{i-k}+B_{i-k}L_{i-k})(B_{i-(j+1)}L_{i-(j+1)}) \cdots \prod_{c=j+2}^{i-1}(I-K_{i-c}C_{i-c})A_{i-c}(I-K_0C_0)\Bigg] - \cdots (B_{i-1}L_{i-1})\prod_{m=2}^{i-1}(I-K_{i-m}C_{i-m})A_{i-m}(I-K_0C_0)$$

π_i is the coefficient of $\mathbf{x}_0$ in x_i. The coefficient of $\mathbf{x}_0$ in y_i is

$$h_{(i+1)1} = C_i[A_i\pi_{i-1} + B_iL_i((I-K_{i-1}C_{i-1}) \cdots (A_{i-1}+B_{i-1}L_{i-1})\theta_{i-2,i-1} - K_{i-1}y_{i-1})]$$

It can be seen that

$$(I-K_{i-1}C_{i-1})(A_{i-1}+B_{i-1}L_{i-1})\theta_{i-2,i-1} - K_{i-1}y_{i-1} = (A_{i-1}+B_{i-1}L_{i-1})j_1 + K_{i-1}C_{i-1}\prod_{b=1}^{i-1}A_{i-b}(I-K_{i-b-1}C_{i-b-1})$$

Hence, the coefficient of $\mathbf{x}_0$ in y_i is

$$h_{(i+1)1} = C_i[A_i\pi_{i-1} + B_iL_i((I-K_{i-1}C_{i-1})(A_{i-1}+B_{i-1}L_{i-1})\theta_{i-2,i-1} - K_{i-1}y_{i-1})] = C_i[A_i\pi_{i-1} + B_iL_i((A_{i-1}+B_{i-1}L_{i-1})\theta_{i-2,i-1} + K_{i-1}C_{i-1}\cdots\prod_{b=1}^{i-1}A_{i-b}(I-K_{i-b-1}C_{i-b-1})$$

The above statement is precisely what we must show. The proof is complete.

Appendix B: Proof of Propositions 2 and 3

Propositions 2 and 3 can be proven by induction. Regarding Proposition 2, for $i = 1, \ldots, j+1$, the proof is obvious. For $i \geq j+2$, we can assume that for $k = a$, $a \geq j+2$, the following statement is true:

$$\tilde{h}_{ij}\mathbf{r}a = C_{i-1}\Bigg[\prod_{m=1}^{i-(j+1)}(A_{i-m}+B_{i-m}L_{i-m})B_jL_j + \sum_{c=1}^{i-(j+2)}\Bigg[\prod_{m=1}^{c}(A_{i-m} + \cdots B_{i-m}L_{i-m})B_{i-(c+1)}L_{i-(c+1)}\prod_{b=c+2}^{i-j}(I-K_{i-b}C_{i-b})A_{i-b}\Bigg] + \cdots B_{i-1}L_{i-1}\prod_{c=2}^{i-j}(I-K_{i-c}C_{i-c})A_{i-c}\Bigg]K_j$$

Then, for $k = a+1$, we can show that

$$\tilde{h}_{(i+1)j} = C_i\Bigg[\prod_{m=1}^{i+1-(j+1)}(A_{i+1-m}+B_{i+1-m}L_{i+1-m})B_jL_j + \cdots \sum_{c=1}^{i+1-(j+2)}\Bigg[\prod_{m=1}^{c}(A_{i+1-m}+B_{i+1-m}L_{i+1-m}) \cdots B_{i+1-(c+1)}L_{i+1-(c+1)}\prod_{b=c+2}^{i+1-j}(I-K_{i+1-b}C_{i+1-b})A_{i+1-b}\Bigg] + \cdots B_{i+1-1}L_{i+1-1}\prod_{c=2}^{i+1-j}(I-K_{i+1-c}C_{i+1-c})A_{i+1-c}\Bigg]K_j$$

References

[1] Mahoney, T. C., and Davis, J., 2017, "Cybersecurity for Manufacturers: Securing the Digitized and Connected Factory," Technical Report.

[2] Langner, R., 2011, "Stuxnet: Dissecting a Cyberwarfare Weapon," IEEE Secur. Priv., **9**(3), pp. 49–51.

[3] Wu, M., 2019, "Intrusion Detection for Cyber-Physical Attacks in Cyber-Manufacturing System," Doctoral Dissertation, Syracuse University, New York.

[4] Lee, R. M., Assante, M. J., and Conway, T., 2014, "German Steel Mill Cyber Attack," Ind. Contr. Syst., **30**(62), pp. 1–15.

[5] Abbaspour, A., Sargolzaei, A., Forouzannezhad, P., Yen, K. K., and Sarwat, A. I., 2019, "Resilient Control Design for Load Frequency Control System Under False Data Injection Attacks," IEEE Trans. Ind. Electron., **67**(9), pp. 7951–7962.

[6] Lu, Y., and Xu, X., 2019, "Cloud-Based Manufacturing Equipment and Big Data Analytics to Enable On-Demand Manufacturing Services," Rob. Comput. Integr. Manuf., **57**, pp. 92–102.

[7] Zhong, R. Y., Newman, S. T., Huang, G. Q., and Lan, S., 2016, "Big Data for Supply Chain Management in the Service and Manufacturing Sectors: Challenges, Opportunities, and Future Perspectives," Comput. Ind. Eng., **101**, pp. 572–591.

[8] Liu, T., Yang, B., Li, Q., Ye, J., Song, W., and Liu, P., 2021, "Cyber-Physical Taint Analysis in Multi-stage Manufacturing Systems (MMS): A Case Study," arXiv preprint.

[9] Shi, J., 2006, *Stream of Variation Modeling and Analysis for Multistage Manufacturing Processes*, CRC Press, Boca Raton, FL.

[10] Al Mamun, A., Liu, C., Kan, C., and Tian, W., 2022, "Securing Cyber-Physical Additive Manufacturing Systems by In-Situ Process Authentication Using Streamline Video Analysis," J. Manuf. Syst., **62**, pp. 429–440.

[11] Shi, Z., Mamun, A. A., Kan, C., Tian, W., and Liu, C., 2022, "An LSTM-Autoencoder Based Online Side Channel Monitoring Approach for Cyber-Physical Attack Detection in Additive Manufacturing," J. Intell. Manuf., **34**, pp. 1815–1831.

[12] Zeltmann, S. E., Gupta, N., Tsoutsos, N. G., Maniatakos, M., Rajendran, J., and Karri, R., 2016, "Manufacturing and Security Challenges in 3D Printing," JOM, **68**(7), pp. 1872–1881.

[13] Liu, C., Kan, C., and Tian, W., 2020, "An Online Side Channel Monitoring Approach for Cyber-Physical Attack Detection of Additive Manufacturing," International Manufacturing Science and Engineering Conference, Vol. 84263, American Society of Mechanical Engineers, p. V002T07A016.

[14] Liu, Y., Ning, P., and Reiter, M. K., 2011, "False Data Injection Attacks Against State Estimation in Electric Power Grids," ACM Trans. Inf. Syst. Security (TISSEC), **14**(1), pp. 1–33.

[15] Northern, B., Burks, T., Hatcher, M., Rogers, M., and Ulybyshev, D., 2021, "VERCASM-CPS: Vulnerability Analysis and Cyber Risk Assessment for Cyber-Physical Systems," Information, **12**(10), p. 408.

[16] Zhang, Y., Jiang, T., Shi, Q., Liu, W., and Huang, S., 2022, "Modeling and Vulnerability Assessment of Cyber Physical System Considering Coupling Characteristics," Int. J. Electr. Power Energy Syst., **142**, p. 108321.
[17] Pan, H., Lian, H., Na, C., and Li, X., 2020, "Modeling and Vulnerability Analysis of Cyber-Physical Power Systems Based on Community Theory," IEEE Syst. J., **14**(3), pp. 3938–3948.
[18] Pivoto, D. G., de Almeida, L. F., da Rosa Righi, R., Rodrigues, J. J., Lugli, A. B., and Alberti, A. M., 2021, "Cyber-Physical Systems Architectures for Industrial Internet of Things Applications in Industry 4.0: A Literature Review," J. Manuf. Syst., **58**, pp. 176–192.
[19] Patan, R., Ghantasala, G. P., Sekaran, R., Gupta, D., and Ramachandran, M., 2020, "Smart Healthcare and Quality of Service in IoT Using Grey Filter Convolutional Based Cyber Physical System," Sustain. Cities Soc., **59**, p. 102141.
[20] Thakur, S., Chakraborty, A., De, R., Kumar, N., and Sarkar, R., 2021, "Intrusion Detection in Cyber-Physical Systems Using a Generic and Domain Specific Deep Autoencoder Model," Comput. Electr. Eng., **91**, p. 107044.
[21] Althobaiti, M. M., Kumar, K. P. M., Gupta, D., Kumar, S., and Mansour, R. F., 2021, "An Intelligent Cognitive Computing Based Intrusion Detection for Industrial Cyber-Physical Systems," Measurement, **186**, p. 110145.
[22] Kazemi, Z., Safavi, A. A., Arefi, M. M., and Naseri, F., 2021, "Finite-Time Secure Dynamic State Estimation for Cyber-Physical Systems Under Unknown Inputs and Sensor Attacks," IEEE Trans. Syst. Man. Cybernet.: Syst., **52**(8), pp. 4950–4959.
[23] Ding, D., Han, Q.-L., Ge, X., and Wang, J., 2020, "Secure State Estimation and Control of Cyber-Physical Systems: A Survey," IEEE Trans. Syst. Man. Cybernet.: Syst., **51**(1), pp. 176–190.
[24] Zhao, Y., Du, X., Zhou, C., and Tian, Y.-C., 2022, "Anti-Saturation Resilient Control of Cyber-Physical Systems Under Actuator Attacks," Inf. Sci., **608**, pp. 1245–1260.
[25] Liao, H.-J., Lin, C.-H. R., Lin, Y.-C., and Tung, K.-Y., 2013, "Intrusion Detection System: A Comprehensive Review," J. Netw. Comput. Appl., **36**(1), pp. 16–24.
[26] Yaacoub, J.-P. A., Salman, O., Noura, H. N., Kaaniche, N., Chehab, A., and Malli, M., 2020, "Cyber-Physical Systems Security: Limitations, Issues and Future Trends," Microprocess. Microsyst., **77**, p. 103201.
[27] Li, D., Gebraeel, N. Z., Paynabar, K., and Meliopoulos, A. S., 2022, "An Online Approach to Covert Attack Detection and Indentification in Power Systems," IEEE Trans. Power Syst., **38**(1), pp. 267–277.
[28] Panigrahi, R., Borah, S., Pramanik, M., Bhoi, A. K., Barsocchi, P., Nayak, S. R., and Alnumay, W., 2022, "Intrusion Detection in Cyber-Physical Environment Using Hybrid Naïve Bayes–Decision Table and Multi-objective Evolutionary Feature Selection," Comput. Commun., **188**, pp. 133–144.
[29] Kwon, H.-Y., Kim, T., and Lee, M.-K., 2022, "Advanced Intrusion Detection Combining Signature-Based and Behavior-Based Detection Methods," Electronics, **11**(6), p. 867.
[30] Song, J., Bandaru, H., He, X., Qiu, Z., and Moon, Y. B., 2020, "Layered Image Collection for Real-Time Defective Inspection in Additive Manufacturing," ASME International Mechanical Engineering Congress and Exposition, Vol. 84492, American Society of Mechanical Engineers, p. V02BT02A006.
[31] Wu, M., Phoha, V. V., Moon, Y. B., and Belman, A. K., 2016, "Detecting Malicious Defects in 3D Printing Process Using Machine Learning and Image Classification," ASME International Mechanical Engineering Congress and Exposition, Vol. 50688, American Society of Mechanical Engineers, p. V014T07A004.
[32] Li, D., Paynabar, K., and Gebraeel, N., 2021, "A Degradation-Based Detection Framework Against Covert Cyberattacks on Scada Systems," IISE Trans., **53**(7), pp. 812–829.
[33] Li, D., Ramanan, P., Gebraeel, N., and Paynabar, K., 2020, "Deep Learning Based Covert Attack Identification for Industrial Control Systems," 2020 19th IEEE International Conference on Machine Learning and Applications (ICMLA), Miami, FL, Dec. 14–17, IEEE, pp. 438–445.
[34] Song, J., Shukla, D., Wu, M., Phoha, V. V., and Moon, Y. B., 2019, "Physical Data Auditing for Attack Detection in Cyber-Manufacturing Systems: Blockchain for Machine Learning Process," ASME International Mechanical Engineering Congress and Exposition, Vol. 59384, American Society of Mechanical Engineers, p. V02BT02A004.
[35] Wu, M., Song, Z., and Moon, Y. B., 2019, "Detecting Cyber-Physical Attacks in Cybermanufacturing Systems With Machine Learning Methods," J. Intell. Manuf., **30**(3), pp. 1111–1123.
[36] Bhardwaj, A., Al-Turjman, F., Kumar, M., Stephan, T., and Mostarda, L., 2020, "Capturing-The-Invisible (CTI): Behavior-Based Attacks Recognition in IoT-Oriented Industrial Control Systems," IEEE Access, **8**, pp. 104956–104966.
[37] Qian, J., Du, X., Chen, B., Qu, B., Zeng, K., and Liu, J., 2020, "Cyber-Physical Integrated Intrusion Detection Scheme in Scada System of Process Manufacturing Industry," IEEE Access, **8**, pp. 147471–147481.
[38] Abokifa, A. A., Haddad, K., Lo, C., and Biswas, P., 2019, "Real-Time Identification of Cyber-Physical Attacks on Water Distribution Systems Via Machine Learning–Based Anomaly Detection Techniques," J. Water Resour. Plann. Manag., **145**(1), p. 04018089.
[39] Urbina, D. I., Giraldo, J. A., Cardenas, A. A., Tippenhauer, N. O., Valente, J., Faisal, M., Ruths, J., Candell, R., and Sandberg, H., 2016, "Limiting the Impact of Stealthy Attacks on Industrial Control Systems," Proceedings of the 2016 ACM SIGSAC Conference on Computer and Communications Security, Vienna Austria, Oct. 24–28, pp. 1092–1105.
[40] Li, D., Gebraeel, N., and Paynabar, K., 2020, "Detection and Differentiation of Replay Attack and Equipment Faults in Scada Systems," IEEE Trans. Autom. Sci. Eng., **18**(4), pp. 1626–1639.
[41] Mo, Y., Chabukswar, R., and Sinopoli, B., 2013, "Detecting Integrity Attacks on Scada Systems," IEEE Trans. Contr. Syst. Technol., **22**(4), pp. 1396–1407.
[42] Van Long, D., Fillatre, L., and Nikiforov, I., 2015, "Sequential Monitoring of Scada Systems Against Cyber/physical Attacks," IFAC-PapersOnLine, **48**(21), pp. 746–753.
[43] Mo, Y., and Sinopoli, B., 2009, "Secure Control Against Replay Attacks," 2009 47th Annual Allerton Conference on Communication, Control, and Computing (Allerton), Monticello, IL, Sept. 30–Oct. 2, IEEE, pp. 911–918.
[44] Li, B., Xiao, G., Lu, R., Deng, R., and Bao, H., 2019, "On Feasibility and Limitations of Detecting False Data Injection Attacks on Power Grid State Estimation Using D-Facts Devices," IEEE Trans. Ind. Inf., **16**(2), pp. 854–864.
[45] Wang, Q., Tai, W., Tang, Y., and Ni, M., 2019, "Review of the False Data Injection Attack Against the Cyber-Physical Power System," IET Cyber-Phys. Syst.: Theory Appl., **4**(2), pp. 101–107.
[46] Jorjani, M., Seifi, H., and Varjani, A. Y., 2020, "A Graph Theory-Based Approach to Detect False Data Injection Attacks in Power System AC State Estimation," IEEE Trans. Ind. Inf., **17**(4), pp. 2465–2475.
[47] Ding, Y., Ceglarek, D., and Shi, J., 2002, "Design Evaluation of Multi-station Assembly Processes by Using State Space Approach," ASME J. Mech. Des., **124**(3), pp. 408–418.