SAGE: Stealthy Attack Generation in Cyber-Physical Systems

Michael Biehler[a], Zhen Zhong[a], and Jianjun Shi[a]*

[a]*H. Milton Stewart School of Industrial and Systems Engineering, Georgia Institute of Technology, Atlanta, USA*

Abstract

Cyber-physical systems (CPS) have been increasingly attacked by hackers. CPS are especially vulnerable to attackers that have full knowledge of the system's configuration. Therefore, novel anomaly detection algorithms in the presence of a knowledgeable adversary need to be developed. However, this research is still in its infancy due to limited attack data availability and test beds. By proposing a holistic attack modeling framework, we aim to show the vulnerability of existing detection algorithms and provide a basis for novel sensor-based cyber-attack detection. Stealthy Attack GEneration (SAGE) for CPS serves as a tool for cyber-risk assessment of existing systems and detection algorithms for practitioners and researchers alike. Stealthy attacks are characterized by malicious injections into the CPS through input, output, or both, which produce bounded changes in the detection residue. By using the SAGE framework, we generate stealthy attacks to achieve three objectives: (i) Maximize damage, (ii) Avoid detection, and (iii) Minimize the attack cost. Additionally, an attacker needs to adhere to the physical principles in a CPS (objective iv). The goal of SAGE is to model worst-case attacks, where we assume limited information asymmetries between attackers and defenders (e.g., insider knowledge of the attacker). Those worst-case attacks are the hardest to detect, but common in practice and allow understanding of the maximum conceivable damage. We propose an efficient solution procedure for the novel SAGE optimization problem. The SAGE framework is illustrated in three case studies. Those case studies serve as modeling guidelines for the development of novel attack detection algorithms and comprehensive cyber-physical risk assessment of CPS. The results show that SAGE attacks can cause severe damage to a CPS, while only changing the input control signals minimally. This avoids detection and keeps the cost of an attack low. This highlights the need for more advanced detection algorithms and novel research in cyber-physical security.

Keywords: Cyber Security; Cyber-Physical Systems; Sensor-based Attack Detection; Stealthy Attack Generation; Anomaly Detection; Critical Infrastructures; Risk Management;

1. Introduction

Cyber-physical attacks are a category of cyber-attacks that also adversely affect the physical space. CPS are characterized by the interaction of physical assets and computational capabilities with information transfer. The rapid digitalization and utilization of CPS lead to the widespread use of sensors, networked devices, and data acquisition systems. Since CPS are deployed for high-value and safety-critical systems, the security of those systems is essential. Any successful attack leads to severe economic loss, equipment damage, or even loss of human life. We find that the limited attack data availability in cyber-physical systems hinders the research on cyber-physical attack detection methods. To develop effective cyber-physical attack detection methods, it is essential to understand the attacker's capabilities and methods. Existing methods to generate an attack utilize random perturbations, which do not integrate the system topology and objectives of an attacker. We find that detection methods using existing types of attack data are not robust to stealthy attacks. This motivates us to develop a general-purpose framework for generating stealthy attacks. Stealthy attacks are characterized by malicious injections into the CPS through input, output, or both, which produce bounded changes in the detection residue.

While stealthy, adversarial attacks have received some attention in the computer vision community, we are the first to holistically integrate the requirements and topology of CPS for the design of stealthy attacks. Attacks in the CPS domain require stealthy attacks beyond image data and the consideration of a wide range of system inputs, models, and tasks (Li et al. 2020).

Therefore, the scope of this paper is to propose a general-purpose optimization framework to find the best strategy to attack CPS and show the implications of such worst-case attacks on existing detection methods. Our framework provides a steppingstone to develop more effective attack detection methods in the future, that are robust to stealthy, worst-case attacks. Worst-case perturbations are defined in terms of the limited information asymmetry between attackers and defenders: While the attacker might not know the specific detection model and its associated parameters, all other data and system information is assumed to be known to the attacker (i.e., insider attacker). This allows us to understand the maximal conceivable damage (i.e., worst-case detection performance) to a CPS for a given system setup and detection strategy.

By formulating a novel optimization problem, the "Stealthy Attack GEneration" (SAGE) framework considers the three main objectives of an attacker (maximize damage, avoid detection, and minimize attack cost) as well as the physical laws in CPS. By applying small, worst-case perturbations to the system input variables, the SAGE attack will lead to unexpected and malicious misbehavior of the system output, while staying undetected by the systems detection algorithms.

To show the generality of our approach, we generate stealthy attacks and validate the SAGE framework on two data modalities: functional curves and image data. For functional curves, we utilize a hot rolling process simulated in MATLAB Simulink. In this setting, we evaluate the performance of seven off-theshelf supervised machine learning models to detect SAGE attacks. In the image case studies, we use the SAGE methods to attack two state-of-the-art methods for image anomaly detection by using a large steel surface defect dataset.

The results provide a case for the severe consequences of stealthy attacks in CPS. This research is intended to serve as a cornerstone for the development of more robust and effective detection algorithms for CPS attacks. Furthermore, by evaluating existing systems and detection models, SAGE can be utilized for the cyber-risk assessment of CPS for practitioners and researchers alike.

The contributions of our SAGE framework are as follows:

- We introduce a comprehensive and general-purpose framework for reliability generating stealthy, worst-case attacks on cyber-physical systems Our model formulation is intuitive and easy to understand, which allows the adaptation to a wide range of cyber-physical systems. We find that many detection methods are unable to detect stealthy SAGE attacks, even when the attacker is oblivious to the specific defense used.
- Our results highlight the need for more comprehensive detection methods: our SAGE framework provides researchers with a common baseline of attack generation, a description of attack techniques, and common evaluation pitfalls, so that future detection methods can avoid falling vulnerable to these same attack procedures.

The remaining parts of this paper are organized as follows: In Section 2, we provide a review of related literature to highlight the necessity of this research. Section 3 presents the mathematical descriptions of CPS, formulates the optimization problem, and proposes an algorithm for solving this problem. In Section 4, we illustrate the methodology with three case studies, which serve as guiding examples for the modeling of stealthy attacks. Finally, Section 5 concludes this paper.

2. Literature Review

Due to the rise of the industrial internet of things (IIoT) and smart manufacturing, CPS have been increasingly exploited by cyber-attacks (Ervural et al. 2018). CPS have grown from stand-alone systems with little security protection to highly interconnected systems that can be easily targeted by attackers over the internet. Attacks like the computer worm “Stuxnet” attacking Siemens industrial software in 2010, or the phishing attack on a German steel mill leading to severe equipment damage in 2014, are some of the most prominent examples of the vulnerability of CPS to cyber-physical attacks. Even though the field of information technology is developing new methodologies for cyber security, the unique characteristics of CPS require specific attention (Zhang et al. 2019).

In general, an attack on a CPS is conducted via three steps. The first step of an attacker is to gain knowledge of the system by identifying the network topology, software, critical targets, and monitoring schemes against cyber-attacks (Han et al. 2014). Then, the attacker needs to bypass the first line of defense consisting of the firewall and an intrusion prevention system. After that, the attacker has full access to the CPS to achieve the goal by perturbing the control systems and making as much damage as possible while staying undetected. This paper focuses on modeling the last step of a CPS attack, which is how to perturb the system inputs to make maximum damage to the system response and stay undetected with minimum cost.

2.1 *Attacks on Cyber-physical Systems*

In the attack domain, attacks on cyber-physical systems can be classified into three main methods: disclosure, disruption, and deception attacks as visualized in Figure 1.

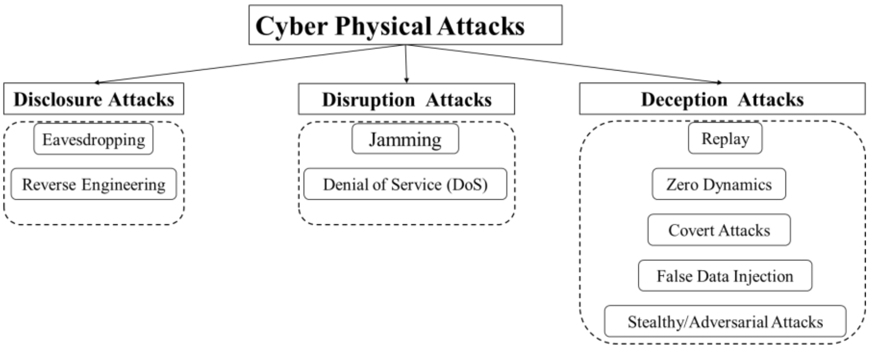

Figure 1: Attack strategies in CPS

Disclosure attacks occur when sensitive or confidential information is exposed to the attacker. Disruption attacks aim at disrupting the physical processes in a CPS. Deception attacks aim to deceive the defender of a system to accept a specific incorrect version of reality (e.g., sensor measurements), causing the defender to act in a way that benefits the attacker (e.g., not raising an alarm). This paper focuses on deception attacks, which can be further classified into five major subtypes of attacks:

- *Replay attack*: The attacker injects a sequence of normal control input into the system using previously recorded sensor data, while actually conducting malicious actions (Murakami et al. 2017).

- *False data injection attack*: The attacker compromises sensor readings in such a way that undetected errors are introduced into the calculation of state variables and values (Ahmed et al. 2020).
- *Zero dynamics attack*: The attacker alters the control output in a way that is consistent with the transmitted control input according to the dynamics of the system (Shim et al. 2022).
- *Covert attack*: The attacker disguises the manipulation of control actions by injecting expected sensor measurements calculated based on the system knowledge (Li et al. 2020).
- *Stealthy/Adversarial attack*: The attacker adds small perturbations to the normal data input. In this way, the detection algorithm in the system will not detect the added perturbation. However, the composed input (normal input + perturbation) will cause a malicious system output and the system output can be precisely determined by the selection of appropriate perturbation by the attacker (Li et al. 2020).

Some of those attack subtypes are highly related and not mutually exclusive. However, stealthy attacks are the hardest ones to detect. They do not solely alter or disguise sensor readings. On the contrary, stealthy attacks add small perturbations to the system control variables. Since those perturbations are so small that there is no need to disguise them. They appear to be caused by the system's natural variability. On the other hand, they will have a detrimental effect on the system outputs. However, the effect on the system output could lead detection algorithms to raise an alarm. Therefore, we will assume that they are disgusted by false data injection or replay attacks or are not monitored.

We note that there are some existing tools for attack generation in CPS (Jeon et al. 2019, Zhang et al. 2021). Those methods carefully design attacks for certain subclasses of cyber-physical systems. With our stealthy attack generation framework, we extend this literature by providing a general-purpose modeling framework, which intuitively integrates all objectives and constraints of an attacker to a CPS and proposes a comprehensive optimization framework to solve this – in general – nonconvex problem to global optimality.

Furthermore, there exist several testbeds and datasets (Conti et al. 2021) for security research in cyber-physical systems in various fields such as electric grids (Hahn et al. 2013) or water treatment plants (Goh et al. 2016). However, those testbeds and datasets might not be well suited for a particular application scenario such as a particular manufacturing system. Therefore, we see the SAGE framework as an extension to a much wider range of systems, which allows vulnerability assessment and robust attack detection development based only on the historic data and the system configuration of the CPS at hand.

2.2 *Machine Learning Methods for CPS Attack Detection*

In recent years, multiple detection algorithms have been developed by utilizing machine learning classifiers for the defense against cyber-attacks (Pasqualetti et al. 2013, Guan et al. 2017, Wu et al. 2019, Yang et al. 2019, Li et al. 2020). Those algorithms have achieved state-of-the-art detection performance on existing types of attack generation schemes such as false data injection, replay, zero dynamics, and even covert attacks. However, those supervised learning techniques require strong assumptions and can be considered as the best-case scenario for the defender of the system: historical training data needs to be available with labels of in-control (e.g., no attack) and attack conditions. Additionally, the current attack needs to come from the same generative process as the historical attacks.

2.3 *Adversarial Machine Learning and Cyber-physical Security*

A large array of prior work has addressed the problem of generating adversarial examples for neural network image classifiers (Akhtar et al. 2018). However, the literature on adversarial data has mainly focused on the image domain, and limited efforts have been made to generalize the concepts to a wide range of data modalities and system models (Zizzo et al. 2019).

Existing works on cyber-physical adversarial attacks are overly specific to one particular system setup or neglect if those attacks are realizable according to the physical laws of the system (Feng et al. 2017, Zizzo et al. 2020). Several methods assume that only a subset of the sensors can be compromised, which tremendously limits the action space for the attacker (Li et al. 2020).

Contrary to adversarial images, the attack generation scheme in CPS needs to consider all three objectives of an attack (e.g., maximize damage, minimize detection, minimize attack cost) and also consider the system model and the physical laws of the system.

2.4 *State-Estimation-based Attacks and Defenses for CPS*

There is a large body of work in state-estimation techniques for cyber-physical intrusion detection in various safety-critical CPS, such as industrial control systems (Inayat et al. 2022) or power grids (Ashok et al. 2016, Guo et al. 2018, Jin et al. 2018). Mo et al. (2012) introduced a framework to generate integrity attacks by formalizing the adversary's strategy as a constrained control problem. However, this method does not consider the physical laws of the system, nor the attack cost. Furthermore, a wide variety of methods have been proposed to attack a CPS by perturbing the state estimation (Kosut et al. 2011, Kim et al. 2014). In a response to those types of threats, robust state estimation techniques have become widespread in practice, nullifying this type of attack scheme (Ding et al. 2020).

In a nutshell, it is essential to investigate the modeling of stealthy attacks for designing more resilient systems and detection algorithms. We will demonstrate that if an attacker knows the current configuration of a CPS, most existing detection algorithms have vulnerabilities and can be bypassed by attackers. Given this fact, the existing attack and detection algorithms are based on too strong assumptions, which may not mimic the behavior of a knowledgeable attacker. Therefore, an effective detection algorithm requires the defenders to first change perspective and "think like a hacker" to identify the weaknesses of a system. By proposing the SAGE framework, we aim to provide a holistic modeling framework that can serve as a stepping stone for the development of more robust attack detection algorithms.

3. SAGE Methodology

This section first describes the system model used to model the dynamics of CPS. Afterward, the SAGE framework is introduced, which considers the main objectives of an attacker consisting of maximizing the damage to the system while staying undetected and keeping the cost of an attack low. Finally, an efficient solution procedure for the nonconvex SAGE formulation is derived.

3.1 *System Modeling*

This section describes the model used to characterize the system dynamics of CPS. For a general CPS, the process outputs $\boldsymbol{Y}_t$ at time t can be in a format of multiple functional curves, images, structured point clouds, or categorical variables. We assume that the effect of the inputs on the outputs can have a hybrid or nonlinear relationship, which allows more realistic modeling of complex CPS. The system model can be obtained with the best fit to the historical data from a variety of potential models like linear regression, gaussian process model, or neural networks, and is represented as

$$\boldsymbol{Y}_t = g_1(\boldsymbol{u}_t, \boldsymbol{\theta_1}) + g_2(\boldsymbol{x}_t, \boldsymbol{\theta_2}) + \boldsymbol{\varepsilon} \qquad (1)$$

where $g_i(\cdot, \boldsymbol{\theta_i})$, $i = 1, 2$ are some general functions (e.g., linear, nonlinear, varying with time) with parameter vector $\boldsymbol{\theta_i}$, representing the effect of the control variables $\boldsymbol{u_t}$ and the process variables $\boldsymbol{x_t}$ (not controllable but observable) on the system output $\boldsymbol{Y}_t$, respectively. $\boldsymbol{\varepsilon}$ is the matrix containing the modeling error where every entry is a zero mean additive Gaussian noise with variance $\boldsymbol{\sigma}^2$. As deep learning approaches are increasingly integrated into CPS, model (1) aims to unify a wide variety of models to model stealthy attacks in nonlinear settings. This general formulation also allows for the hybrid settings of linearized and nonlinear perception pipelines that are fused in a deterministic or stochastic manner.

3.2 *Stealthy Attack GEneration (SAGE) formulation*

This subsection will first discuss the threat model, which is a structured representation of all the information and assumptions that affect the security of a CPS. Afterward, the SAGE attack formulation and solution procedures are presented.

3.2.1 *Threat model*

To model the worst-case scenario for a defender, we assume that the attacker knows the system configuration in a gray box setting. In particular, it is assumed that the attacker has bypassed the first line of defense (i.e., the firewall) and has full access to the system. Thus, the attacker can inject control actions at any point and time. It is assumed that an attacker has an intention to negatively affect the system output. Examples of such damage to the system include a reduction in the production rate, production quality, system efficiency, equipment degradation, or failure. We assume that the attacker has full knowledge of the system model. In the white box setting, the attacker directly knows the model g_i and its parameters $\boldsymbol{\theta_i}$. In the gray box setting, the attacker estimates a surrogate model for g_i based on historical data. For the detection algorithm used in the CPS system, we assume a black-box or gray-box setting: (i) If the attacker does not know the systems detection algorithm (black box), they can generically minimize the difference in distribution between normal and attack data as illustrated in the steel rolling case study in Section 4.1. (ii) In the gray box setting, we assume that the attacker knows the detection algorithm, but not the specific detection model parameters. In this setting, an attacker can only estimate the detection model parameters based on historical data. The attacker does not know the specific out-of-control or attack data utilized during model training by the defender. We assume that all the systems control and process variables are being monitored. The system output measurements are either (i) disguised through false data injection or covert attacks, (ii) not monitored, or (iii) monitored far downstream in a multi-stage (manufacturing) system, which already would have caused severe upstream damage until its detection.

We note that this threat model restricts the attackers' capabilities as little as possible. Therefore, it is extremely stealthy and hard to detect.

3.2.2 *Attackers optimization problem – "Think like a hacker"*

Based on the threat model, which summarizes the attackers' capabilities, we will "think like a hacker" (Esteves et al. 2017) and define three key attacker's objectives when generating stealthy attacks on a CPS:

i. *Maximize Damage:* The goal of an attacker is to cause damage to physical components such as machines, equipment, parts, assemblies, and products in CPS. Thus, the cyber attacker can cause severe damage to CPS by increasing the wear, breakage, scrap, or any other changes to the original design.

ii. *Avoid Detection:* An attacker aims to manipulate CPS in such a way that the altered control actions stay undetected. Most equipment has some hard-wired safety modes that will shut down the machines once they reach a safety-relevant operating condition. Therefore, staying undetected will directly contribute to the first objective to maximize damage.

iii. *Minimize Attack Cost:* Attacking all control actions might be costly or complicated because different sensing data are saved in different databases or governed by different operating systems or security protocols. Therefore, the attacker will want to keep the cost of an attack low by identifying very few control actions that have the biggest impact on the system outputs.

iv. *Physical limits*: Any changes to the system need to adhere to the *physical limits* of the CPS.

Consequently, the attacker’s optimization problem is formulated as Equation 2, which exploits the CPS system model and the weaknesses of the detection algorithm while considering the physical constraints of the system.

$$\min_{u_t^A} - \left\| d\left(g_1(\boldsymbol{u}_t^A, \boldsymbol{\theta}_1) + g_2(x_t^{IC}, \boldsymbol{\theta}_2)\right) - d\left(g_1(u_t^{IC}, \boldsymbol{\theta}_1) + g_2(x_t^{IC}, \boldsymbol{\theta}_2)\right)\right\|_p \tag{2a}$$

$s.t.$

$$\left\| f(\boldsymbol{u}_t^{IC}) - f(\boldsymbol{u}_t^A)\right\|_p \leq \varepsilon_1 \tag{2b}$$

$$C(\boldsymbol{u}_t^A) \leq \varepsilon_{2,} \tag{2c}$$

$$\left\| ph(\boldsymbol{u}_t^A)\right\|_p \leq \varepsilon_3, \tag{2d}$$

where $d(\cdot)$ denotes a damage function corresponding to some undesirable outputs of a system given the in-control and attack-control actions, respectively. Furthermore, $\boldsymbol{u}_t^A$ are the perturbed control inputs by the attacker, which should be close to the normal or in-control control inputs $\boldsymbol{u}_t^{IC}$. The process variables, which are not controllable, are denoted by $\boldsymbol{x}_t^{IC}$**.** The distances are denoted in terms of the ℓ_p-norm to allow for flexible modeling requirements. ε_1 denotes the maximal allowable distance (i.e., decision boundary) between some general detection or monitoring function $f(\cdot)$ applied to $\boldsymbol{u}_t^A$ and $\boldsymbol{u}_t^{IC}$; ε_2 denotes the maximal allowable cost of an attack strategy $\boldsymbol{u}_t^A$, and ε_3 denotes the maximal allowable range from the physical laws modeled by a general function $ph(\cdot)$ of the attack.

The detailed explanation of each term in Equation 2 is as follows:

- The objective function (2a) incorporates the first objective of the attacker, which is to *maximize the damage* to the system. This is equivalent to minimizing the negative difference between the damage function $d(\cdot)$ for the in-control and the attacker’s control actions respectively. If only the system output deviation is of concern, $d(\cdot)$ reduces to the identity function. In cases where the state space has significant asymmetries, the $d(\cdot)$ functions can be defined as a (binary) mapping to a dangerous state. Note, that the process variables will cancel in this formulation since those are not controllable and therefore should be kept at their in-control values during the attack.
- The 1st constraint (2b) term corresponds to the second objective of the attacker, which is to *avoid detection*. A detection algorithm is represented by a general function $f(\cdot)$. By ensuring the ℓ_p-norm distance between the output of the detection function $f(\cdot)$ applied to both attackers and in-control control actions falls below the detection threshold ε_1, the attacker can avoid detection.
- The 2nd constraint (2c) term corresponds to the last objective of an attacker, which is to *minimize the attack cost*. This term considers how costly it is to attack a particular control action. The executed changes to the control variables should be within the attacker’s (computational) budget

ε_2. Examples of increased attack costs could be cases in which different control subsystems are secured by different mechanisms (e.g., firewalls) with different levels of security or the computational effort to execute changes to control variables is high.

- The last constraint (2d) term ensures that the *physical limits* of the CPS are met via a physics function $ph(\cdot)$. The function $ph(\cdot)$ maps the attacker's actions to the physical constraints. Control actions can only change within physical limits ε_3 (e.g., the magnitude of change in consecutive time steps should be small). This term requires physical knowledge of the process, which can be obtained from domain experts or prior research findings.

The system model in Equation (1) is known in advance or at least the predictions are accessible in a black box manner. The functions $f(\cdot)$ and $ph(\cdot)$ are also known in advance. In Table 1, several common monitoring statistics and physical constraints are introduced as guiding examples for the choice of $f(\cdot)$ and $ph(\cdot)$. If the monitoring scheme or physical constraints are not known, the functions can be chosen as the identity and variance function by default as introduced in the steel rolling case study in Section 4.1. To further enhance this strategy, a distributional distance such as the Kullback-Leibler or Wasserstein could be selected as the monitoring function $f(\cdot)$. Without knowing any particular details about the applied detection model, this approach is still able to fool common machine learning classifiers as illustrated in the case study.

Table 1: Modeling examples for monitoring function $f(\cdot)$ and physical constraint $ph(\cdot)$

Monitoring Scheme	$f(\cdot)$	**Physical Constraint**	$ph(\cdot)$
X-bar & S Charts	Identity + Variance	Smooth changes over time	$u_{ij,t}^{A} - u_{ij,t-1}^{A}$
Hotelling T^2 Control Chart	T^2 statistic	Sparse changes over time	$\left\| u_{ij,t}^{A} - u_{ij,t-1}^{A} \right\|_1$
Kernel Methods (e.g., SVM or PCA)	Corresponding Kernel function	Limited variation patterns	$\left\| u_{ij,t}^{A} - u_{ij,t-1}^{A} \right\|_*$ $\|\cdot\|_*$ *denotes the nuclear norm*
Gradient Boosting	Weighted sum of weak learners	Piecewise constant changes	Fused lasso penalty (Tibshirani et al. 2005)
Neural Network Architectures	Inverse network function via back-propagation	Variables within physically possible limits	$\left\| u_{ij,t}^{A} \right\|_2^2$ with appropriate Lagrange multiplier λ_2

It should be pointed out that the SAGE attack is designed for patient adversaries, who have collected historic data about the process, or insider attackers. Therefore, the SAGE attacker is an expert of the CPS to be attacked. Thus, it is reasonable to assume that the system model $g_1(\cdot)$ and $g_2(\cdot)$, the monitoring function $f(\cdot)$, and physical limits $ph(\cdot)$ are known in advance before conducting the optimization for stealthy attack control actions. This assumption is reasonable since CPS use industry-wide standards in terms of control systems and even detection algorithms. This assumption does not artificially limit the capabilities of an attacker. It rather leads to extremely stealthy attacks, which may be considered the worst-case scenario for the defender. However, if a defender constantly changes the monitoring algorithm or even the system setup, SAGE attacks may not be able to simultaneously fulfill their four objectives. Also, in cases when the detection model that monitors the system is not fully characterized, this attack framework might not lead to stealthy attacks. Another limitation is the possibly nonconvex formulation, which has an optimality guarantee only under certain conditions as discussed in the next section.

Using the Karush-Kuhn-Tucker (KKT) conditions, we can reformulate Equation 2 to alleviate the burden of explicitly computing inequality constraints as follows:

$$\min_{u_t^A} - \left\| d\left(g_1(\boldsymbol{u}_t^A, \boldsymbol{\theta_1}) + g_2(x_t^{IC}, \boldsymbol{\theta_2})\right) - d\left(g_1(u_t^{IC}, \boldsymbol{\theta_1}) + g_2(x_t^{IC}, \boldsymbol{\theta_2})\right)\right\|_p$$
$$+\lambda_1 \left\| f(\boldsymbol{u}_t^{IC}) - f(\boldsymbol{u}_t^A)\right\|_p + \lambda_2 C(\boldsymbol{u}_t^A) + \lambda_3 \left\| p(\boldsymbol{u}_t^A)\right\|_p \tag{3}$$

where λ_1, λ_2 and λ_3 denote the Lagrange multipliers that correspond to the constraints (2b), (2c) and (2d), respectively.

The global minimum of the original constrained optimization problem (Equation 2) corresponds to a saddle point in the Lagrangian function (Equation 3), provided that the necessary regularity conditions of stationarity, primal feasibility, dual feasibility, and complementary slackness are satisfied. For a more detailed explanation of this widely used approach, interested readers are referred to (Ben-Tal et al. 2001). We note that for nonconvex optimization problems, the Lagrange multipliers λ_1, λ_2 and λ_3 may not be unique. Therefore, we resort simultaneously solving for the optimal solution and the appropriate Lagrange multipliers by utilizing the Branch-and-Reduce framework introduced in Subsection 3.2.3.

3.2.3 *Solution Procedure*

The SAGE problem formulation is an inherently nonconvex and NP-hard problem. To make the SAGE framework applicable to a wide range of general nonconvex functions, the Branch-And-Reduce Optimization Navigator (BARON) algorithm is utilized to solve the nonconvex formulation to global optimum (Liu et al. 2019).

The output dimension of the nonconvex constraint functions is denoted by m_1, m_2 and m_3, respectively, and $\boldsymbol{X}$ denotes a set of constraints for the search space. For example, $\boldsymbol{X}$ could denote the 6σ limits of the attacked control variables, because any attack outside of those limits can very easily be detected. The standard Lagrangian subproblem of Equation 2 is given in Equation 3. However, for the dual approach to yield any computational advantage, the so-called Lagrangian subproblem must be much easier to solve than the primal problem.

For simplicity of the problem presented in the remaining paper, we use the following notations to replace the related terms in Equation 2:

$$\begin{aligned}
&\boldsymbol{x} = \boldsymbol{u}_t^A \in \mathbb{R}^n,\\
&o_1(\boldsymbol{x}) = -\left\| d\left(g_1(\boldsymbol{u}_t^A, \boldsymbol{\theta_1}) + g_2(x_t^{IC}, \boldsymbol{\theta_2})\right) - d\left(g_1(u_t^{IC}, \boldsymbol{\theta_1}) + g_2(x_t^{IC}, \boldsymbol{\theta_2})\right)\right\|_p : \mathbb{R}^n \to \mathbb{R},\\
&o_2(\boldsymbol{x}) = \left\|f(\boldsymbol{u}_t^{IC}) - f(\boldsymbol{u}_t^A)\right\|_p \leq \varepsilon_1 \colon \mathbb{R}^n \to \mathbb{R}^{m_1},\\
&o_3(\boldsymbol{x}) = C(\boldsymbol{u}_t^A) \leq \varepsilon_2 \colon \mathbb{R}^n \to \mathbb{R}^{m_2},\\
&o_4(\boldsymbol{x}) = \left\|ph(\boldsymbol{u}_t^A)\right\|_p \leq \varepsilon_3 \colon \mathbb{R}^n \to \mathbb{R}^{m_3}
\end{aligned}$$

Then, Equation 2 can be defined as the Lagrangian subproblem:

$$\inf_{\boldsymbol{x}\in \boldsymbol{X}} l'(\boldsymbol{x}, (\lambda_0, \lambda_1, \lambda_2, \lambda_3)) = \inf_{x\in X}\{-\lambda_0 o_1(\boldsymbol{x}) - \lambda_1 o_2(\boldsymbol{x}) - \lambda_2 o_3(\boldsymbol{x}) - \lambda_3 o_4(\boldsymbol{x})\}, \tag{4}$$

where $(\lambda_0, \lambda_1, \lambda_2, \lambda_3) \leq 0$. The additional dual variable λ_0 homogenizes the problem and allows us to reformulate the SAGE attack into a unified BARON range-reduction problem. The constraints $\varepsilon_1, \varepsilon_2$ and ε_3 enter the Lagrangian subproblem as $\lambda_1\varepsilon_1, \lambda_2\varepsilon_2$, and $\lambda_3\varepsilon_3$, respectively. Therefore, they are constants that do not alter the optimal solution and only need to be considered in the Lagrangian master problem (Equation 3). Assume that b_0 is an upper bound on the optimal objective function value of Equation 2 and consider the following range-reduction problem:

$$\begin{aligned}
h^* = \inf_{x,u_0,u_1,u_2,u_3} &\{h(u_o, u_1, u_2, u_3) | o_1(\boldsymbol{x}) \leq u_0 \leq b_0,\\
&o_2(\boldsymbol{x}) \leq u_1 \leq \varepsilon_1, o_3(\boldsymbol{x}) \leq u_2 \leq \varepsilon_2, o_4(\boldsymbol{x}) \leq u_3 \leq \varepsilon_3,\\
&\boldsymbol{x} \in \boldsymbol{X}\},
\end{aligned} \tag{5}$$

where h is assumed to be some semi-continuous functions. Then, Equation 5 can be restated as

$$h^* = \inf_{x,u_0,u_1,u_2,u_3} h(u_0, u_1, u_2, u_3)$$

$s.t.$

$$\begin{gathered}
-\lambda_0(o_1(\boldsymbol{x}) - u_0) - \lambda_1(o_2(\boldsymbol{x}) - u_1) - \lambda_2(o_3(\boldsymbol{x}) - u_2) - \lambda_3(o_4(\boldsymbol{x}) - u_3) \leq 0\\
(\lambda_0, \lambda_1, \lambda_2, \lambda_3) \leq 0\\
(u_0, u_1, u_2, u_3) \leq (b_0, \varepsilon_1, \varepsilon_2, \varepsilon_3)\\
\boldsymbol{x} \in \boldsymbol{X}
\end{gathered} \tag{6}$$

However, the computational complexity of Equation 6 is the same as Equation 4. Therefore, we lower bound h^* with the optimal value of the following problem.

$$h_L = \inf_{x, u_0, u_1, u_2, u_3} h(u_0, u_1, u_2, u_3)$$

$s.t.$

$$\lambda_0 u_0 + \lambda_1 u_1 + \lambda_2 u_2 + \lambda_3 u_3$$
$$+ \inf_{x \in X} \{-\lambda_0 o_1(\boldsymbol{x}) - \lambda_1 o_2(\boldsymbol{x}) - \lambda_2 o_3(\boldsymbol{x}) - \lambda_3 o_4(\boldsymbol{x})\} \leq 0$$
$$(\lambda_0, \lambda_1, \lambda_2, \lambda_3) \leq 0$$
$$(u_0, u_1, u_2, u_3) \leq (b_0, \varepsilon_1, \varepsilon_2, \varepsilon_3) \tag{7}$$

This domain reduction problem can be leveraged for efficiently solving the SAGE attack by restricting $h(u_0, u_1, u_2, u_3)$ to $a_0 u_0 + a_1 u_1 + a_2 u_2 + a_3 u_3$, where $(a_0, a_1, a_2, a_3) \geq 0$ and $(a_0, a_1, a_2, a_3) \neq 0$. Using Fenchel-Rockafellar duality, the BARON algorithm derived in (Tawarmalani 2001) can be applied to iteratively obtain lower and upper bounds on the range-reduction problem of the SAGE attack formulation.

Branch- and Reduce (BARON) algorithm to solve the SAGE attack

```
While not converged:
(0) Initialize: Set $K = 0, u_0^0 = a_0, u_1^0 = \varepsilon_1, u_2^0 = \varepsilon_2, u_3^0 = \varepsilon_3$
(1) Solve the relaxed dual of Equation 5:
        $h_U^K = \max_{u_0,u_1,u_2,u_3} (\lambda_0 + a_0)b_0 + (\lambda_1 + a_1)\varepsilon_1 + (\lambda_2 + a_2)\varepsilon_2 + (\lambda_3 + a_3)\varepsilon_3 - z$
    $s.t.$  $z \geq \lambda_0 u_0^k + \lambda_1 u_1^k + \lambda_2 u_2^k + \lambda_2 u_2^k, k = 0, \dots, K-1$
            $(\lambda_0, \lambda_1, \lambda_2, \lambda_3) \leq -(a_0, a_1, a_2, a_3)$
    Let the solution be $(\lambda_0^K, \lambda_1^K, \lambda_2^K, \lambda_3^K)$
(2) Solve the Lagrangian subproblem:
        $\inf_{x, u_0,u_1,u_2,u_3} l'(\boldsymbol{x}, (\lambda_0^K, \lambda_1^K, \lambda_2^K, \lambda_3^K)) = -\max_{x,u_0,u_1,u_2,u_3} \lambda_0^K u_0 + \lambda_1^K u_1 + \lambda_2^K u_2 + \lambda_3^K u_3$
            $s.t.$  $o_1(\boldsymbol{x}) \leq u_0$
                    $o_2(\boldsymbol{x}) \leq u_1$
                    $o_3(\boldsymbol{x}) \leq u_2$
                    $o_4(\boldsymbol{x}) \leq u_3$
                    $\boldsymbol{x} \in \boldsymbol{X}$
    Let the solution be $(\boldsymbol{x}^K, u_0^K, u_1^K, u_2^K, u_3^K)$.
(3) Augment and solve the relaxed primal problem:
        $h_L^K = \min_{u_0,u_1,u_2,u_3} a_0 u_0 + a_1 u_1 + a_2 u_2 + a_3 u_3$
            $s.t.$  $\lambda_0^k u_0 + \lambda_1^k u_1 + \lambda_2^k u_2 + \lambda_3^k u_3$
            $+ \inf_{\boldsymbol{x} \in \boldsymbol{X}} l'(\boldsymbol{x}, (\lambda_0^k, \lambda_1^k, \lambda_2^k, \lambda_3^k)) \leq 0, k = 1, \dots, K$
        $(u_0, u_1, u_2, u_3) \leq (b_0, \varepsilon_1, \varepsilon_2, \varepsilon_3)$
4) Termination check:
    If $h_U^K - h_L^K \leq tolerance$
```

In step 2 of the algorithm, $(\lambda_0, \lambda_1, \lambda_2, \lambda_3) \leq 0$ implies that $u_0^K = f(\boldsymbol{x}^K), u_1^K = g_1(\boldsymbol{x}^K)$, $u_2^K = g_2(\boldsymbol{x}^K)$, and $u_3^K = g_3(\boldsymbol{x}^K)$. Furthermore, the relaxations of the BARON framework enjoy quadratic convergence properties and are an efficient procedure for obtaining global optima to nonlinear programs (Tawarmalani et al. 2004). In particular, the theorem for optimality-based range reduction (Tawarmalani 2001) applies to the derived BARON algorithm for solving the SAGE attack:

Theorem 1 (Tawarmalani 2001). Suppose the Lagrangian subproblem in Equation 5 is solved for certain dual multipliers $(\lambda_0, \lambda_1, \lambda_2, \lambda_3) \leq 0$. Then, for each i such that $(\lambda_0^i, \lambda_1^i, \lambda_2^i, \lambda_3^i) \neq 0$, the cuts $g_p^i(\boldsymbol{x}) \geq (b_0 - \inf_{\boldsymbol{x}} l(\boldsymbol{x}, \lambda_0, \lambda_1, \lambda_2, \lambda_3)/\lambda_p^i,\ p = 0{,}1{,}2{,}3)$ do not chop off any optimal solution of the initial Equation 4.

This theorem implies that the solution will eventually converge to a global optimum due to the quadratic convergence of the BARON algorithm. For a detailed discussion, related proofs, and generalizations we refer interested readers to (Tawarmalani 2001).

The BARON algorithm to solve the SAGE formulation is also available as commercial software (Sahinidis 1996). For readers interested in generating a SAGE attack with no in-depth optimization knowledge or no commercial nonlinear solver licenses, we recommend solving the SAGE formulation using efficient and widely used algorithms such as stochastic gradient descent (SGD). In the literature, several convergence guarantees are provided for SGD algorithms in the nonconvex setting (Nguyen et al. 2018). When using SGD on common software platforms, a few best practices should be considered. The attacks should be initialized with historic, in-control data. This will lead to much faster convergence. Furthermore, choosing upper and lower bounds within the physical limits of the data (e.g., image pixel values from 0 to 255, control variables within 6σ limits) will reduce the probability of detection and drastically reduce the solution space of the problem. The choice of Lagrange multipliers of the SAGE formulation is crucial to the efficacy of the attack. Binary search can be adapted to find the optimal set of parameters for any arbitrary choice of algorithm. The binary search should consider the three main objectives of the attacker and tune the hyperparameters $\lambda_l, l = 1{,}2{,}3$ until the Attack Effectivity (AE), Average Perturbation (AP), and Attack Cost (AC) are within prescribed limits. The attack effectivity can either be computed by the first SAGE term or by an attack-specific metric considering the attacked system model. Similarly, the average perturbation can be derived from terms (2b) and (2d) or the defender's monitoring algorithm. The attack cost is directly calculated from the third SAGE term.

4. Case Studies

In this section, we use three case studies to illustrate and validate the SAGE methodology proposed in Section 3. Those case studies are intended as modeling guidelines for the application to other CPS. We will demonstrate how to use the proposed framework for two data modalities: functional curve and image data. All the case studies follow the same SAGE framework proposed in Equation 2. However, the formulations need to be adapted to the specific case. To summarize the procedure, we provide a pseudo-code with the respective inputs and outputs of each case study. The case studies are intended to serve as guiding examples for the generation of stealthy attacks in a wide range of systems.

```
Pseudo-Code SAGE Attack procedure
Inputs:
    • Historic data of normal (in-control) control actions: Section 4.1: $u_t^{IC}$; Section 4.2: $y_t^{original}$, Section 4.3: $\theta_t^{original}, \xi^{original}$
    • System model ($g_1$ and $g_2$ with parameters $\theta_1$ and $\theta_2$ in Eq. 2): Section 4.1: $B_0, \beta_j$; Section 4.2: $\theta_\alpha^{SSD}$; Section 4.3: $\theta^{CNN}, \xi^{LIME}$
    • Detection function or statistic ($f$ in Eq. 2): Section 4.1: Multivariate EWMA statistic, Section 4.2 and 4.3: Identity function
    • Cost function: $C$ in Eq. 2 and Section 4.1-4.3
    • Damage function ($d$ in Eq. 2): Sections 4.1-4.3: Identity function
    • Physical constraint function ($ph$ in Eq. 2): Section 4.1: Temporal consistency $u_t^A - u_{t-1}^A$; Section 4.2: Temporal consistency $y_{t-1}^A - y_t^A$; Section 4.3: Spatial-temporal consistency $\mathrm{vec}\left(y_i^{original} - y_i^A\right) \cdot \mathcal{D} \cdot \mathrm{vec}\left(y_i^{original} - y_i^A\right)^T$
    • Reference value or historic data of system output (: Section 4.1: $Y_t^{ref}$; Section 4.2: $\theta_\alpha$; Section 4.3: $y_t^{original}$
    • Threshold values: Detection threshold $\varepsilon_1$, Attack cost limit $\varepsilon_{2,}$ Physical limit $\varepsilon_3$
Outputs: Attackers control actions ($u_t^A$ in Eq. 2): Section 4.1: $u_t^A$; Section 4.2: $y_t^A$; Section 4.3: $y_t^A$
SAGE Attack procedure:
    1. Specify the objective function in the format of Eq. 2
    2. Solve for the attacker's control action utilizing the Branch-and-Reduce framework introduced in Section 3.2.3
    3. Deploy the attacker's control actions to the cyber-physical system
```

4.1 *Case Study with Functional Curve Data – Hot Steel Rolling Process*

To show the vulnerability of common CPS to stealthy attacks, a MATLAB Simulink testbed (MathWorks 2022) for one-stage plate rolling is used to illustrate the devasting effect of small but worst-case perturbations on functional curves in CPS. The testbed models a two-axis rolling mill. In a rolling process, steel rollers are used to press sheet metal to a specific thickness and add strength via strain hardening to improve surface finish. The four control inputs to this system are the roller gap and roller force in x-direction and y-direction, respectively. The testbed uses a Multiple Input Multiple Output (MIMO) LQG regulator to control the horizontal and vertical thickness of a steel plate in a hot steel rolling mill. For further details on the testbed setup, interested readers are referred to the corresponding Simulink documentation (MathWorks 2022). The only modification to the testbed is the addition of four "import" blocks to link the attacker's control signals generated from the SAGE formulation to the simulated CPS as shown in Figure 2. The import blocks allow us to verify the accuracy of the system model. We will report the actual output signal obtained from the simulation after injecting the attackers' control actions. This is more accurate and realistic than simply plugging the attacker's control actions back into a data-driven system model, which is only an approximation of the true system dynamics.

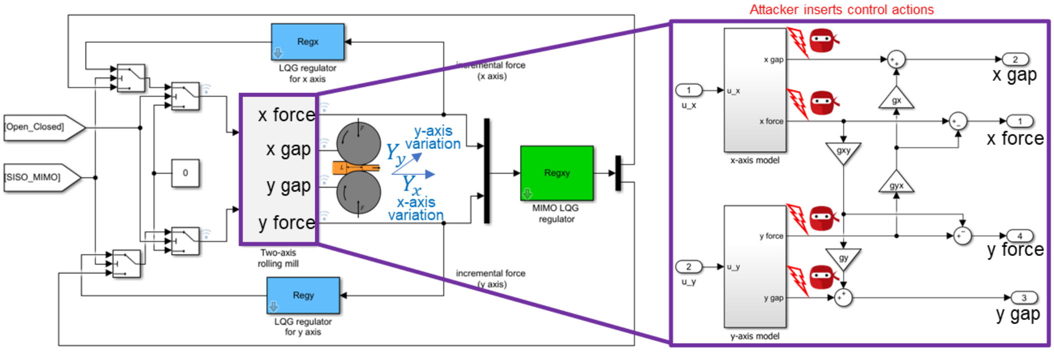

Figure 2: SAGE attack to a hot rolling process

The monitoring of multivariate signals via multi-variate control charts is a standard practice in the industry and also in some recent research papers. Therefore, we use the first case study to illustrate the potential of SAGE to deceive a very popular detection model based on a multivariate exponentially weighted moving average (MEWMA) control chart.

4.1.1 Attack on Multivariate-EWMA Chart

A multivariate exponentially weighted moving average (MEWMA) control chart is commonly used to monitor CPS. We first calculate $Z_i = \Lambda X_i + (1 - \Lambda)\mathrm{Z}_{\mathrm{i}-1}$, where Z_t is the i-th observation vector, Z_0 is the vector of variable values from historic data, Λ is the $diag(\lambda_1, \lambda_2, \ldots \lambda_p)$, which is a diagonal matrix with $\lambda_1, \lambda_2, \ldots \lambda_p \in (0,1]$ on the main diagonal, and p is the number of control variables. Then the test statistics of the MEWMA is given by $T_i^2 = Z_i^T \sum_{Z_i}^{-1} Z_i$. The alarm will be triggered whenever T_i^2 is above the 1-α=95% quantile of its empirical distribution under normal conditions, and α is the desired Type-I error rate. We incorporate this test statics directly into the framework by minimizing the monitoring static on the attacker's control to avoid detection.

$$\min_{\boldsymbol{u}_t^A} -\left\|\boldsymbol{Y}_t^{ref} - \boldsymbol{B}_0 - \sum_{j=1}^{4} \beta_j u_{j,t}^A\right\|_2^2 + \lambda_1 \left\|T_t^2(\boldsymbol{u}_t^A)\right\|_2^2 + \lambda_2 \left\|\boldsymbol{u}_t^A - \boldsymbol{u}_{t-1}^A\right\| + \lambda_3 C(\boldsymbol{u}_t^A), \qquad (9)$$

where $\boldsymbol{Y}_t^{ref} = [Y_x, Y_y]$ denotes the engineering specification of quality response and is a constant value in this case. The four control variables are denoted by $\boldsymbol{u} = [\mathrm{x_{force}}, \mathrm{x_{gap}}, \mathrm{y_{force}}, \mathrm{y_{gap}}]$. In this setting, $d\big(g_1(\boldsymbol{u}_t) + g_2(\boldsymbol{x}_t)\big) = d(g_1(\boldsymbol{u}_t) + \boldsymbol{B}_0) = \boldsymbol{Y}_t^{ref}$ and $d(\cdot)$ reduces to the identity function (i.e., $d(\cdot) = Id(\cdot)$). Since the system response, in this case, is measured in terms of x- and y-axis thickness variation, the goal would be to have no variation so $\boldsymbol{Y}_t^{ref} = \vec{0}$. In this case, $\boldsymbol{u}_t^{IC}$ is chosen as historic data of the same length as the attack to mimic a replay attack. Furthermore, the cost function is chosen as $C(u_{j,t}^{A})=\begin{cases}0, \ for\, j = 1,3 \\ 2, for\, j = 2,4\end{cases}$. This represents the fact that the roll gap ($j = 1,3$) is easy to attack while the roller force ($j = 2,4$) requires more effort because they are protected through different security protocols. The monitoring statistic $f(\cdot)$ is set to the MEWMA monitoring statistic T_t^2.

The optimization problem was solved using the proposed BARON algorithm for the SAGE formulation in Section 3.3.2. For better visualization, only 150-time steps of the attack are visualized in the following figures. The attack avoids detection by the MEWMA chart as visualized in Figure 3.

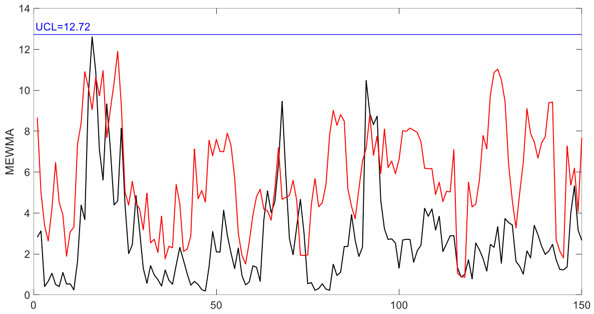

Figure 3: Attackers' control actions (red) and in-control data (black) are both within the control limits of the MEWMA chart

On the other hand, the attack leads to maximal damage to the system response, which is far away from the normal system response (Figure 4).

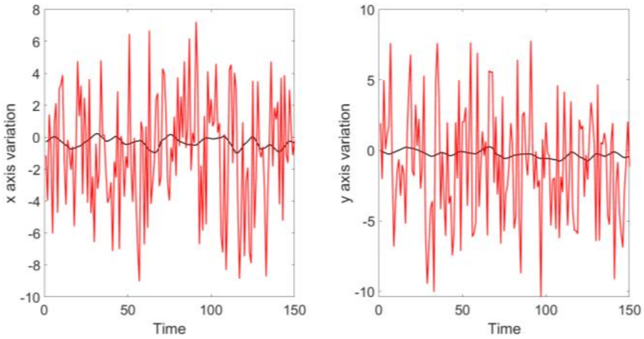

Figure 4: System response after an attack (red) and in-control system response (black)

4.1.2 SAGE attack performance evaluation and comparison with other methods

To show that small perturbations of the control variables can lead to a large change in the system response, the Attack Effectivity (AE) and Average Perturbation (AP) are computed as follows.

- Attack effectivity $AE = \frac{\sum_{j=1}^{4}\left(\sum_{t=1}^{n}\left\|u_{j,t}^{IC} - u_{j,t}^{A}\right\|/n\right)}{\left(\sum_{j=1}^{4}\sigma_{u_j^{IC}}\right)}$
- Average Perturbation $AP = \frac{\sum_{t=1}^{n}\left\|\boldsymbol{Y}_t^{ref} - \boldsymbol{Y}_t^{A}\right\|/n}{\boldsymbol{\sigma}_Y}$,

where n denotes the length of the attack, $\sigma_{u_j^{IC}}$ is the in-control standard deviation of control variable j, $\boldsymbol{\sigma}_Y$ is the in-control standard deviation of the system responses, and $\boldsymbol{Y}_t^A$ is the resulting system response to the attack. Those metrics essentially measure the absolute distance between in-control and attack in terms of the number of in-control standard deviations. The results are summarized in Table 2 showing the small perturbation levels of the attacks while achieving very effective attacks.

Table 2: Attack Effectivity and Average Perturbation of SAGE attacks

	AE	AP
MEWMA Attack	11.024	0.123

To further evaluate the effectiveness of the proposed SAGE attack, seven machine learning techniques commonly used in literature for cyber-attack detection algorithms in CPS are evaluated for their effectiveness to detect stealthy attacks (Table 5). The hyperparameters of the respective methods were tuned via grid search to achieve the best possible detection results. In particular, a Support Vector Machine (SVM), k Nearest Neighbor (kNN), Random Forest (RF), Bagging, Gradient Boosting Machine (GBM), Decision Tree (DT), and a Deep Neural Network (DNN) were used to classify the presence of an attack. The labels for those supervised machine learning methods are obtained as follows: The normal operating conditions are labeled as no attack, and the generated attack signals obtained via our SAGE framework are labeled as an attack. This shows the potential of our method: using our stealthy attack framework, sophisticated attacks can be generated, which can be utilized for supervised learning approaches.

We note that the proposed SAGE attacks were not aware of those detection algorithms when we formalized the SAGE optimization problem. In particular, those detection functions were not considered as a detection function $f(\cdot)$ during the optimization of the attack. We simply enforce the MEWMA, which will incapacitate most of the detection methods. Note that not even distributional distances such as Kullback-Leibler or Wasserstein distances had to be utilized to fool those detection methods.

The results in Table 3 show that if the SAGE attack considers the MEWMA statistic, which ensures that the attack and in-control data are similar in terms of their first two distribution moments (mean and (co)variance), none of those six methods can achieve satisfactory detection performance. While the DNN performs the best, its detection accuracy of 54.79% is not sufficient for reliable and fast attack detection. Note that a random coin flip (i.e., attack, no attack) at each time point would result in a 50% accuracy.

Table 3: Detection results of different machine learning methods (**bold** marks best-performing)

Method	MEWMA Attack			
	Accuracy	Precision	Recall	F1-Score
SVM	48.28%	50.50%	47.64%	49.03%
kNN	47.89%	49.75%	46.89%	48.28%
RF	48.58%	49.53%	48.61%	49.07%
Bagging	48.49%	51.04%	46.89%	48.88%
GBM	51.84%	52.11%	51.83%	51.97%
DT	52.08%	52.36%	51.81%	52.08%
DNN	**54.79%**	**54.05%**	**55.56%**	**54.79%**

This example shows how flexible the SAGE formulation can be adjusted to make the existing detection algorithms not effective even if the detection algorithm such as the machine learning classifiers are not known *a priori.*

4.2 *Case Study with Image Data*

In this subsection, we will provide a generalization of the SAGE attack to learning-enabled CPS utilizing two state-of-the-art anomaly detection algorithms. Another goal of this case study is to illustrate the potential of the SAGE framework on other data formats, in particular image data. We provide a case for the severe consequences of small but intentional perturbations to control variables on image responses in CPS. Therefore, we will attack both the smooth spare decomposition (SSD) method (Yan et al. 2017), which is a benchmark image denoising and anomaly detection algorithm in the field of manufacturing, and a Convolutional Neural Network in combination with Local Interpretable Model-Agnostic Explanations (Ribeiro et al. 2016), which is a state-of-the-art method in the field of classification and object detection.

The dataset used for both attacks is the Northeastern University (NEU) surface defect database (Song et al. 2013), which contains six typical surface defects of hot-rolled steel strips. The dataset includes 1,800 grayscale images, with 300 samples of each of the six different surface defects (i.e., rolled-in scale (RS), patches (Pa), crazing (Cr), pitted surface (PS), inclusion (In) and scratches (Sc)).

4.2.1 *SAGE Attack on Smooth-Sparse-Decomposition*

Firstly, we attack the SSD method (Yan et al. 2017), which decomposes an image into three components: A smooth image background, sparse anomalous regions, and random noise, as illustrated in Figure 5.

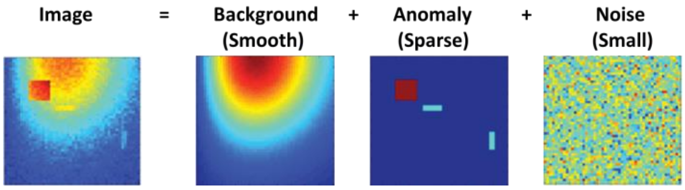

Figure 5: Decomposition of the image into the background, anomaly, and noise (Yan et al. 2017)

The goal of the attack is to add small perturbations to the image, which are indistinguishable from the original image for the human eye. To make this perceived image loss more objective, we measure the distance between normal and attacked images in terms of a L_2-norm perturbation, which is a standard procedure in computer vision.

However, those perturbed attack images should lead to a bad system response. In this case, the system response is the anomaly region. We want to change the anomaly region as much as possible. When decomposing the image into background, anomaly, and noise via SSD, we want to detect the anomalies in different regions than where they actually are. This means, when the operators try to fix the problem, they will draw a wrong conclusion regarding the root cause of the anomalies and make the damage even worse by taking the wrong actions. In this circumstance, the SAGE attack formulation reduces to the following optimization problem.

$$\min_{\boldsymbol{y}_t^A} - \left\|\boldsymbol{\theta}_\alpha - \boldsymbol{\theta}_\alpha^{SSD}(\boldsymbol{y}_t^A)\right\|_F^2 + \lambda_1 \left\|\boldsymbol{y}_t^{original} - \boldsymbol{y}_t^A\right\|_2 + \lambda_2 \left\|\boldsymbol{y}_t^{original} - \boldsymbol{y}_t^A\right\|_1 + \lambda_3 \left\|\boldsymbol{y}_{t-1}^A - \boldsymbol{y}_t^A\right\|_2^2 \quad (11)$$

where $\boldsymbol{y}_t^A$ denotes the image that the attacker will inject into the system at time t, $\boldsymbol{\theta}_\alpha$ denotes the fixed and known anomaly region of the normal image, $\boldsymbol{\theta}_\alpha^{SSD}$ is a function of $\boldsymbol{y}_t^A$ and denotes the extracted anomaly region from the attacker's image via the SSD method. The goal of the attacker is to maximize the damage by letting $\boldsymbol{\theta}_\alpha^{SSD}$ be as far away as possible from the ground truth anomaly $\boldsymbol{\theta}_\alpha$. Furthermore, to avoid being detected, the attackers' image $\boldsymbol{y}_t^A$ should be close to the original image before the attack $\boldsymbol{y}_t^{original}$ in terms of a L_2-norm perturbation. The computational cost increases with the number of pixels attacked in an image. Therefore, the cost function is chosen as the l_1-norm to induce sparsity and attack as few pixels as possible. Since the monitoring of a process usually consists of streaming data from each time step *t*, the added perturbations in consecutive time steps should not be too different since this might be physically impossible. Furthermore, extreme changes over time might alert appropriate detection algorithms and lead to detection. This behavior is enforced by the 3rd and the last terms in the formulation (Equation 11).

The SAGE attack on SSD (Equation 11) was solved using the BARON framework introduced in Section 3.2.3. As shown in Figure 6, the image before and after the attack is almost indistinguishable to the human eye.

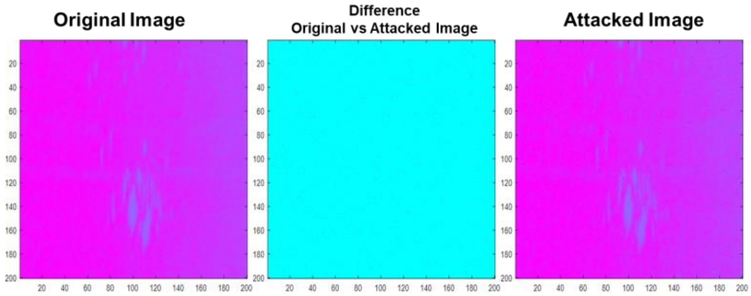

Figure 6: Images before and after the attack of exemplary steel surface defect

On the other hand, the outputs of the SSD algorithm before and after the attack are significantly different (Figure 7). After the attack, the false alarm rate has increased significantly since many regions are now identified incorrectly as surface defects. This effect cannot be achieved by simply adding random noise to the images since the SSD method inherently decomposes the pictures in a smooth, sparse, and noise component.

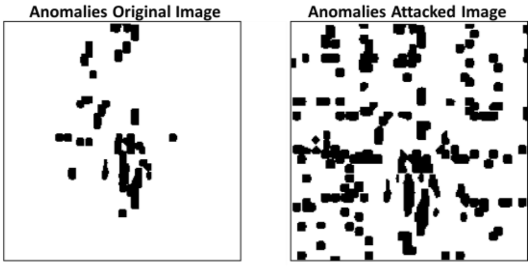

Figure 7: Exemplary recovered anomaly using SSD from the original and the attacked images

To show the generality of the SAGE formulation in attacking multiple classes of anomalies, the entire data set of 1,800 images is selected and the following metrics are defined corresponding to the objectives of the attacker.

- Attack effectivity: $AE = \frac{\sum \mathbf{1}_{>0}\left(\left|\boldsymbol{\theta}_{\alpha}^{original} - \boldsymbol{\theta}_{\alpha}^{A}\right|\right)}{\mathbf{1}_{>0}(\boldsymbol{\theta}_{\alpha}^{A})}$

- Average Pixel Perturbation: $APP = \frac{\sum_{k=1}^{n}\sum_{l=1}^{m}\left|Y_{kl}^{original} - Y_{kl}^{A}\right|}{n \cdot m \cdot 255}$,

where n and m denote the height and width of the image, respectively. In this case study, the images have the size $n = m = 200$. The larger the attack effectivity (AE), the more damage the attacker can do to the anomaly region; and the smaller the average pixel perturbation (APP), the closer the attacked image will be to the original image. Note that the APP is scaled by 255 to account for the range of the pixel intensity values from 0 to 255. The averaged results of those metrics for the 1,800 images are shown in Table 4.

Table 4: Attack effectivity and Average Pixel Perturbation of SAGE attack applied to SSD

	AEE	APP
SAGE Attack	40.534%	0.0482

As we can see from the results of the surface defects, after applying small but intentional perturbations via the SAGE framework, the SSD algorithm can be fooled by falsely adding and/or deleting anomaly regions, while generating an attack image that is virtually indistinguishable for the human eye. This case study shows the generality of the SAGE framework when applied to image data even for sophisticated anomaly detection algorithms like SSD, which utilizes advanced optimization techniques. Therefore, our proposed framework can easily be adapted for other image anomaly detection methods as long as the parameters of the detection algorithms are explicitly known or at least predictions from the detection algorithm can be accessed in a black box manner.

4.2.2 *SAGE Attack on CNN-LIME*

This case study will use the SAGE strategy to attack a CNN-LIME (Ribeiro et al. 2016). Local Interpretable Model-Agnostic Explanations (LIME) explain the prediction of any classifier by treating it as a black box model and learning an interpretable model locally around the prediction. LIME finds the region of an image that leads to the classification of that image to a particular class. Given this fact, it is related to object detection algorithms that locate objects of interest in an image by predicting a boundary around the object. Based on previous research, object detection algorithms are much more difficult to attack (Xie et al. 2017). Therefore, attacking CNN-LIME will demonstrate the immense capabilities of the proposed SAGE formulation in attacking a wide range of algorithms.

4.2.2.1 Development of a CNN-LIME model based on the NEU surface detection datasets

In the modeling efforts, transfer learning with weights from the MobileNet is utilized to obtain a good classification model. A 99.9% model accuracy is achieved by initializing the CNN architecture with those weights and fine-tuning it on the NEU surface detection dataset. These accuracy results utilizing transfer learning outperform recently published results from (He et al. 2019) on a ResNet50 trained from scratch on the dataset (99.67% accuracy). Therefore, the results can be considered state-of-the-art performance on the NEU dataset. Afterward, the LIME algorithm is utilized to explain the predictions of the CNN model and identify the anomaly regions in the images. Let the CNN model be denoted by $f: \mathbb{R}^d \to \mathbb{R}$, where $f(y)$ is the probability that an image y belongs to a certain class. Furthermore, $\Pi_y(z)$ denotes the proximity measure or locality between an image z to y. Lastly, $\mathcal{L}(f, g, \Pi_y)$ measures how unfaithful g is in approximating f in the locality defined by Π_y. To ensure both interpretability and local fidelity, the explanation produced by LIME is obtained by balancing $\mathcal{L}(f, g, \Pi_y)$ and $\Omega(g)$, which is a measure of complexity (as opposed to interpretability) of the explanation g via the following optimization problem.

$$\boldsymbol{\xi}(\boldsymbol{y}) = argmin_g \mathcal{L}(f, g, \Pi_y) + \Omega(g) \tag{12}$$

LIME has achieved state-of-the-art explanatory performance of CNN classification results on critical applications such as tumor classification. Interested readers are referred to the results in (Ribeiro et al. 2016) for further details and links to the corresponding code repository.

4.2.2.2 SAGE attack on the CNN-LIME

The SAGE attack for the CNN-LIME is formalized as the following:

$$\begin{aligned}\min_{\boldsymbol{y}_t^A} - \left\|\boldsymbol{\theta}_t^{original} - \boldsymbol{\theta}^{CNN}(\boldsymbol{y}_t^A)\right\|_F^2 - \lambda_0 \left\|\boldsymbol{\xi}^{original} - \boldsymbol{\xi}^{LIME}(\boldsymbol{y}_t^A)\right\|_F^2 + \lambda_{1,1}\left\|\boldsymbol{y}_t^{original} - \boldsymbol{y}_t^A\right\|_2 \\ + \lambda_{1,2}\left\|\text{vec}\left(\boldsymbol{y}_i^{original} - \boldsymbol{y}_i^A\right) \cdot \mathcal{D} \cdot \text{vec}\left(\boldsymbol{y}_i^{original} - \boldsymbol{y}_i^A\right)^T\right\|_F^2 + \lambda_2 \left\|\boldsymbol{y}_t^{original} - \boldsymbol{y}_t^A\right\|_1\end{aligned} \tag{13}$$

where $\boldsymbol{\theta}$ denotes the predicted class probabilities and $\boldsymbol{\xi}$ is the explanation produced by LIME for the class predictions. "Maximizing damage" in this setting consists of two parts: firstly, the attacker aims to misclassify the anomaly images, and secondly the attacker aims to change the explanatory region away from the original one to make the attacker's malicious class prediction seems legitimate. To avoid detection, we minimize the L_2-norm perturbation between the original and the attacker's image $\boldsymbol{y}_t^A$.

Furthermore, the changes in the image should be smooth to preserve the spatial dependencies to avoid detection. Therefore, the smoothness penalty $\lambda_{1,2} \left\| \mathrm{vec}\left(\boldsymbol{y}_t^{original} - \boldsymbol{y}_t^A\right) \cdot \boldsymbol{\mathcal{D}} \cdot \mathrm{vec}\left(\boldsymbol{y}_t^{original} - \boldsymbol{y}_t^A\right)^T \right\|_F^2$ is applied, where $\boldsymbol{\mathcal{D}}$ is the second-order smoother that applies to the vectorized difference between the original and the attacker's image. Additionally, the increase in computational cost with each attacked pixel is penalized via a L_1-norm sparsity constraint. Similar to the image attack on the SSD algorithm, the attacker's image can hardly be distinguished from the original one as shown in Figure 8.

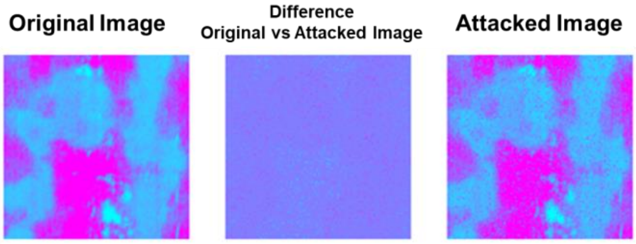

Figure 8: Original image (left), added perturbations (middle), and attackers' image (right) of exemplary surface defect

In this attack formulation, the goal is to misclassify a given true process anomaly class as any of the remaining five class labels. From the results in Table 7, we can see that the correct class patches (Pa) are identified with very high confidence (99.6%) before the attack. After the attack, the probability of the correct class reduces to 0.8%, and the class inclusion (In) was chosen with the highest confidence (69.5%). It can also be observed that the exemplary classification result changes significantly among different faulty patterns in the NEU data sets as shown in Table 5.

Table 5: Exemplary Classification Results before and after the attack (highest-class probability **bold**)

Class Label	RS	PS	Cr	Pa	In	Sc
Before Attack	0.001	0.000	0.002	**0.996**	0.001	0.001
After Attack	0.041	0.107	0.021	0.0836	**0.6954**	0.052

Any other process anomaly class can be attacked similarly as summarized for the 1,800 images in the dataset in Table 8. If the attacker not only wants to misclassify the anomalies but also assigns the picture to a specific prescribed class, the first penalty term in Equation 12 can be adjusted accordingly.

The second term of the 1st objective of the attack (i.e., maximize damage) was to change the explanatory region derived via LIME as far as possible from the original one to avoid any suspicion and justify the differently classified anomaly after the SAGE attack on the image. Figure 9 shows an example of the severe change in the explanatory region after the attack.

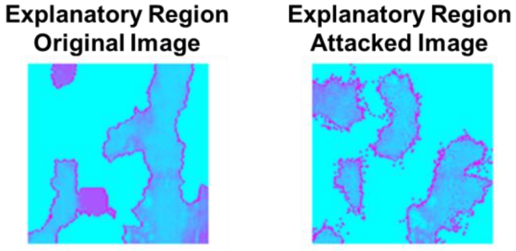

Figure 9: Original explanatory region computed via LIME (left) and attacked explanatory region (right) of exemplary surface defect

The small pixels around the identified regions after the attack coincided with the inclusion anomaly, which has the highest-class probability after the attack. This will avoid detection by the defender while leading to wrong conclusions about the underlying process anomaly.

The SAGE attack on CNN-LIME was applied to the entire dataset of 1,800 images. The evaluation metrics for those attacks are as follows:

The change of classification is denoted as the Ratio of Attacked to Clean correct class Accuracy (RACA) as follows: $RACA = \frac{1}{n}\frac{\sum_{i\in Attack} L(\boldsymbol{y}_i^A|\boldsymbol{Y}_i)}{\sum_{i\in Original} L(\boldsymbol{y}_i^{original}|\boldsymbol{Y}_i)}$, where $L()$ denotes the accuracy loss of a single picture $\boldsymbol{y}_i$ with true class $\boldsymbol{Y}_i$ and n is the number of image samples. Note, a smaller score (RACA) indicates a better attack. The change in the LIME explanatory region is denoted by the attack effectivity (AE) as defined earlier. The attacker's perturbation to the input image is denoted by the average pixel perturbation (APP) as defined earlier. The averaged results for the entire dataset are reported in Table 6.

Table 6: Average Attack Metrics of the SAGE Attack applied to CNN-LIME

	RACA for CNN	AE for LIME	APP
SAGE Attack	22.495%	69.534%	0.0716

The results show the significant effectiveness of the general SAGE attack on a large number of image classification results computed via CNN-LIME. We note that the SSD algorithm is much more vulnerable to perturbations than CNN-LIME. The SSD attacks exploit very few weak spots in the image and change the pixel value significantly to destroy the smoothness of the background. The CNN-LIME attack has a slightly higher APP of 0.0716. To both change, the classification result and explanatory region, a much larger number of pixels need to be attacked. However, the SAGE formulation can exploit the weaknesses of both SSD and CNN-LIME very effectively. Because of this fact, the SAGE attack provides an effective generalization for existing adversarial example generation schemes in the setting of a black-box attack.

Even in the case of black-box attacks, where the detection algorithm is not known to the attacker, the proposed SAGE framework can cause severe damage to a system while staying undetected by commonly used machine-learning classifiers. This provides a strong case for the generality and effectiveness of the proposed framework. SAGE can not only exploit weaknesses of particular algorithms through its flexible formulation but also make replay non-essential for effective attacks by mimicking normal operating conditions.

5. Conclusion

We have introduced SAGE as a holistic framework for attack generation in CPS, which incorporates the three main objectives of an attacker (maximize damage, avoid detection, and minimize the attack cost) and the physical constraints of the CPS. This research is intended as a stepping stone for researchers to develop new research methodologies for cyber-physical attack detection.

The results of this paper make a case that by solving the proposed optimization problem, SAGE attacks can have devastating effects on CPS while staying undetected by system monitoring algorithms. This directly highlights the urgent need for further research in the detection methodology that studies the stealthy and adversarial behavior of cyber-physical attacks. By proposing an efficient algorithm with convergence guarantees for solving this nonconvex optimization problem, we provide a comprehensive modeling platform for stealthy attacks on CPS. We compare our SAGE framework with several main-streams attack detection techniques, which did not utilize stealthy attacks as their input data. We show that the performance deteriorates significantly under worst-case, stealthy attacks.

The SAGE framework can also be used to evaluate newly developed detection algorithms: By plugging the detection function back into the second objective of the attacker (avoid detection), the robustness of cyber-physical attack detection algorithms can be evaluated: If the detection performance degrades below a certain threshold (e.g., 50% corresponding to a random guess), it is an indication that the proposed algorithm is not robust towards stealthy attacks. As an intermediate sanity check, we suggest black-box attacks for a given system as illustrated in the hot-steel rolling case study: In this setting, we did not directly specify the detection algorithm during the SAGE attack generation. The attack data was just regularized to mimic the normal operating conditions in terms of the EWMA statistic. Newly developed detection schemes should have state-of-the-art performance on such types of benchmark attacks.

The limitations of the proposed framework are settings, in which a large number of detection methods are used to monitor the systems: In this setting, it becomes hard to find a globally optimal solution, that maximizes damage, avoids detection, keeps the attack cost low and stays within the physical limits of the system. However, in this setting, the false-alarm rate also might be inflated under normal operating conditions. Furthermore, SAGE currently is not able to dynamically adjust to changes in the detection method. Our future work will address those two limitations.

Acknowledgments

This research is funded by National Science Foundation Award ID 2019378.

References

Ahmed, M. and Pathan, A.-S. (2020) False Data Injection Attack (FDIA): An overview and new metrics for fair evaluation of its countermeasure. *Complex Adaptive Systems Modeling* **8**(1): 1-14.

Akhtar, N. and Mian, A. (2018). Threat of Adversarial Attacks on Deep Learning in Computer Vision: A survey. *IEEE Access* **6**: 14410-14430.

Ashok, A., Govindarasu, M. and Ajjarapu, V. (2016). Online detection of stealthy false data injection attacks in power system state estimation. *IEEE Transactions on Smart Grid* **9**(3): 1636-1646.

Ben-Tal, A. and Nemirovski, A. (2001). Lectures on modern convex optimization: analysis, algorithms, and engineering applications, *SIAM*.

Conti, M., Donadel, D. and Turrin, F. (2021). A survey on industrial control system testbeds and datasets for security research. *IEEE Communications Surveys & Tutorials* **23**(4): 2248-2294.

Ding, D., Han, Q.-L., Ge, X., and Wang, J. (2020). Secure state estimation and control of cyber-physical systems: A survey. *IEEE Transactions on Systems, Man, and Cybernetics: Systems* **51**(1): 176-190.

Ervural, B. C. and Ervural, B. (2018). *Overview of cyber security in the industry 4.0 era. Industry 4.0: managing the digital transformation*, Springer, Cham

Esteves, J., Ramalho, E., and De Haro, G. (2017). To improve cybersecurity, think like a hacker. *MIT Sloan Management Review* **58**(3): 71.

Feng, C., Li, T., Zhu, Z., and Chana, D. (2017). A deep learning-based framework for conducting stealthy attacks in industrial control systems. *arXiv preprint arXiv:1709.06397*.

Goh, J., Adepu, S., Junejo, K. N., and Mathur, A. (2016). A dataset to support research in the design of secure water treatment systems. *International Conference on Critical Information Infrastructures Security*, Springer.

Guan, Y. and Ge, X. (2017). Distributed attack detection and secure estimation of networked cyber-physical systems against false data injection attacks and jamming attacks. *IEEE Transactions on Signal and Information Processing over Networks* **4**(1): 48-59.

Guo, Z., Shi, D., Johansson, K. H., and Shi, L. (2018). Worst-case stealthy innovation-based linear attack on remote state estimation. *Automatica* **89**: 117-124.

Hahn, A., Ashok,A., Sridhar, S., and Govindarasu, M. (2013). Cyber-physical security testbeds: Architecture, application, and evaluation for smart grid. *IEEE Transactions on Smart Grid* **4**(2): 847-855.

Han, S., Xie, M., Chen, H.-H., and Ling, Y. (2014). Intrusion detection in cyber-physical systems: Techniques and challenges. *IEEE Systems Journal* **8**(4): 1052-1062.

He, Y., Song, K., Meng, Q., and Yan, Y. (2019). An end-to-end steel surface defect detection approach via fusing multiple hierarchical features. *IEEE Transactions on Instrumentation and Measurement* **69**(4): 1493-1504.

Jeon, H. and Eun, Y. (2019). A stealthy sensor attack for uncertain cyber-physical systems. *IEEE Internet of Things Journal* **6**(4): 6345-6352.

Jin, M., Lavaei, J., and Johansson, K. H. (2018). Power grid AC-based state estimation: Vulnerability analysis against cyber attacks. *IEEE Transactions on Automatic Control* **64**(5): 1784-1799.

Kim, J., Tong, L. and Thomas, R. J. (2014). Data framing attack on state estimation. *IEEE Journal on Selected Areas in Communications* **32**(7): 1460-1470.

Kosut, O., Jia, L., Thomas, R. J., and Tong, L. (2011). Malicious data attacks on the smart grid. *IEEE Transactions on Smart Grid* **2**(4): 645-658.

Li, D., Paynabar, K. and Gebraeel, N. (2020). A degradation-based detection framework against covert cyberattacks on SCADA systems. *IISE Transactions*: 1-18.

Li, F., Li, Q., Zhang, J., Kou, J., Ye, J., Song, W., and Mantooth, H. A. (2020). Detection and Diagnosis of Data Integrity Attacks in Solar Farms Based on Multilayer Long Short-Term Memory Network. *IEEE Transactions on Power Electronics* **36**(3): 2495-2498.

Li, J., Lee, J. Y., Yang, Y., Sun, J. S., and Tomsovic, K. (2020). Conaml: Constrained adversarial machine learning for cyber-physical systems. *arXiv preprint arXiv:2003.05631*.

Li, J., Liu, Y., Chen, T., Xiao, Z., Li, Z., and Wang, J. (2020). Adversarial attacks and defenses on cyber–physical systems: A survey. *IEEE Internet of Things Journal* **7**(6): 5103-5115.

Liu, J., Ploskas, N., and Sahinidis, N. V. (2019). Tuning BARON using derivative-free optimization algorithms. *Journal of Global Optimization* **74**(4): 611-637.

MathWorks. (2022). MATLAB Simulink : Beam Thickness Control., retrieved from https://www.mathworks.com/help/control/ug/thickness-control-for-a-steel-beam.html.

Mo, Y. and Sinopoli, B. (2012). Integrity attacks on cyber-physical systems. *Proceedings of the 1st International Conference on High Confidence Networked Systems.*

Murakami, K., Suemitsu, H. and Matsuo, T. (2017). Classification of repeated replay-attacks and its detection monitor. *2017 IEEE 6th Global Conference on Consumer Electronics (GCCE)*

Nguyen, L., Nguyen, P. H., Dijk, M., Richtárik, P., Scheinberg, K., and Takác, M. (2018). SGD and Hogwild! convergence without the bounded gradients assumption. *International Conference on Machine Learning*, PMLR.

Pasqualetti, F., Dörfler, F., and Bullo, F. (2013). Attack detection and identification in cyber-physical systems. *IEEE Transactions on Automatic Control* **58**(11): 2715-2729.

Ribeiro, M. T., Singh, S., and Guestrin, C. (2016). " Why should I trust you?" Explaining the predictions of any classifier. *Proceedings of the 22nd ACM SIGKDD International Conference on Knowledge Discovery and Data Mining*.

Sahinidis, N. V. (1996). BARON: A general purpose global optimization software package. *Journal of Global Optimization* **8**(2): 201-205.

Shim, H., Back, J., Eun, Y., Park, G., and Kim, J. (2022). *Zero-dynamics Attack, Variations, and Countermeasures. Security and Resilience of Control Systems*, Springer, Cham

Song, K. and Yan, Y.(2013). A noise robust method based on completed local binary patterns for hot-rolled steel strip surface defects. *Applied Surface Science* **285**: 858-864.

Tawarmalani, M. (2001). Mixed Integer Nonlinear Programs: Theory, Algorithms and Applications, University of Illinois at Urbana-Champaign.

Tawarmalani, M. and Sahinidis, N. V. (2004). Global optimization of mixed-integer nonlinear programs: A theoretical and computational study. *Mathematical Programming* **99**(3): 563-591.

Tibshirani, R., Saunders, M., Rosset, S., Zhu, J., and Knight, K. (2005). Sparsity and Smoothness via the Fused Lasso. *Journal of the Royal Statistical Society: Series B (Statistical Methodology)* **67**(1): 91-108.

Wu, M., Song, Z., and Moon, Y. B. (2019). Detecting cyber-physical attacks in CyberManufacturing systems with machine learning methods. *Journal of Intelligent Manufacturing* **30**(3): 1111-1123.

Xie, C., Wang, J., Zhang, Z., Zhou, Y., Xie, L., and Yuille, A. (2017). Adversarial examples for semantic segmentation and object detection. *Proceedings of the IEEE International Conference on Computer Vision.*

Yan, H., Paynabar,K. and Shi, J. (2017). Anomaly detection in images with smooth background via smooth-sparse decomposition. *Technometrics* **59**(1): 102-114.

Yang, B., Guo, L., Li, F., Ye, J., and Song, W., (2019). Vulnerability assessments of electric drive systems due to sensor data integrity attacks. *IEEE Transactions on Industrial Informatics* **16**(5): 3301-3310.

Zhang, F., Kodituwakku, H., Hines, J. W., and Coble, J. (2019). Multilayer data-driven cyber-attack detection system for industrial control systems based on network, system, and process data. *IEEE Transactions on Industrial Informatics* **15**(7): 4362-4369.

Zhang, K., Keliris, C., Parisini, T., and Polycarpou, M. (2021). Stealthy Integrity Attacks for a Class of Nonlinear Cyber-Physical Systems. *IEEE Transactions on Automatic Control.*

Zizzo, G., Hankin, C., Maffeis, S., and Jones, K. (2019). Adversarial machine learning beyond the image domain. *2019 56th ACM/IEEE Design Automation Conference (DAC)*

Zizzo, G., Hankin, C., Maffeis, S., and Jones, K. (2020). Adversarial attacks on time-series intrusion detection for industrial control systems. *2020 IEEE 19th International Conference on Trust, Security and Privacy in Computing and Communications (TrustCom)*