Algorithm Design for Resilient Cyber-Physical Systems using an Automated Attack Generative Model

SUBMISSION DATE / POSTED DATE

17-11-2021 / 29-11-2021

CITATION

Zheng, Yu; Sayghe, Ali; Anubi, Olugbenga (2021): Algorithm Design for Resilient Cyber-Physical Systems using an Automated Attack Generative Model. TechRxiv. Preprint.
https://doi.org/10.36227/techrxiv.17032898.v1

DOI

10.36227/techrxiv.17032898.v1

Algorithm Design for Resilient Cyber-Physical Systems using an Automated Attack Generative Model

Yu Zheng, *Student Member, IEEE,* , Ali Sayghe, *Student Member, IEEE,* and Olugbenga Moses Anubi, *Senior Member, IEEE,*

***Abstract*—This paper presents a suite of algorithms for detecting and localizing attacks in cyber-physical systems, and performing improved resilient state estimation through a pruning algorithm. High performance rates for the underlying detection and localization algorithms are achieved by generating training data that cover large region of the attack space. An unsupervised generative model trained by physics-based discriminators is designed to generate successful false data injection attacks. Then the generated adversarial examples are used to train a multi-class deep neural network which detects and localizes the attacks on measurements. Next, a pruning algorithm is included to improve the precision of localization result and provide performance guarantees for the resulting resilient observer. The performance of the proposed method is validated using the numerical simulation of a water distribution cyber-physical system.**

***Index Terms*—Resilient Control and Estimation, False Data Injection Attack, Generative Model, Multilayer Perceptron, Pruning algorithm**

I. Introduction

Modern cyber-physical critical infrastructures (CPCI) are facing fast-evolving cyber threats. Cyberattacks can strongly impact the operation of the CPCIs. For instance, suspected cyber intruder took control of the Prykarpattyaoblenergo power system control center in western Ukraine in December 2015, leaving 230,000 people without electricity for up to 6 hours. That was the first time hackers have successfully targeted a country's power grid [1]. Also, cyber attacks in water systems have already become a reality. In 2015, 25 cyber attacks were disclosed in several water systems [2]. And recently in 2020, a malicious cyber-attack attempted to raise the chlorine level in Israel's water supply to dangerous proportion [3]. Supervisory Control and Data Acquisition (SCADA) is a critical part of the CPCI that is highly susceptible to cyberattacks. SCADA is responsible for collecting measurements from Remote Terminal Units (RTU) and sending them to the Control Centers (CC) to perform various functions such as contingency analysis, optimal planning, state estimation (SE), etc. The primary purpose of those functions is to maintain stable and secure operation of the CPCI.

SE is considered a core function in the CCs, that regularly performs in real-time to monitor the system states using information collected from the SCADA systems and other measurement devices. SE can also detect and identify bad data that can cause a significant error in the resulting estimate. The SE algorithm's accuracy depends on how pure and accurate the system measurements are. If any malicious or erroneous measurements pass through undetected, it could mislead the operators into making catastrophically wrong decisions.

Numerous studies have shown that SE is vulnerable to False Data Injection (FDI) Attack, where an intruder aims to hack multiple RTUs or even communication channels to insert fake measurements to misguide the CCs operational process [4]–[7]. Since FDI attack was introduced in [4], different researchers have proposed various techniques for detection and localization, including diagnostic robust generalized potential [8], generalized likelihood ratio [9], fast Go-Decomposition (GoDec) [10], Markov chain [11], Bayesian detection with binary hypothesis [12], and cosine similarity matching scheme [13] and some residual-based approaches such as the Kullback Leibler distance method [14], unscented Kalman filter [15], χ^2 failure detector [16] and a residual-based localization scheme [17]. However, due to their dependency on the system model, the associated model uncertainty would affect their accuracy, and if the FDI attacks are designed properly, these model-based or residual-based detectors can easily be bypassed [4], [17].

To overcome the limitations of traditional residual-based bad data detection approaches, data-driven solutions based on machine learning algorithms have been widely adopted for detecting and localizing of FDI attacks due to their fast execution times and accurate results [18]–[24]. [18] utilized and tested various machine learning algorithms in detecting FDI attacks. The results showed that machine learning algorithms can detect FDI attacks accurately and faster. [19] proposed a deep neural network algorithm that can automatically detect and localize FDI attacks in the power system. The proposed

[1]The authors are with Department of Electrical and Computer Engineering, FAMU-FSU College of Engineering, Center for Advanced Power Systems, Florida State University, Tallahassee, FL 32310 {yz19b, asayghe, oanubi}@fsu.edu

Acknowledgement: This material is based upon work supported by the Department of Energy under Award Number DE-CR0000005

method can integrate the underlying graph topology of the grid and spatial correlations of its measurement data to jointly detect and and localize the FDI attacks in power systems. [24] proposed a hierarchy of neural networks with capability of detecting novel attacks by training networks to understand the entire normal space instead of attack space.

In addition to attack detection and localization, resilient estimation approaches focus on maintaining the correctness of state estimates in the presence of adversarial attacks, which enables resilient operation of the CPSs [25]. Although attacks are possibly unbounded and can be designed to "fool" ℓ_2 observers [4], [16], the attack signals are sparse due to resource limitation of the attackers. Based on this characteristics, the authors in [26] proposed an ℓ_1 minimization program for estimator design on linear systems and proposed an upper-bound on the total number of compromised measurements for guaranteed secure estimation. Some extension work considered robustness on resilient ℓ_1 estimation in presence of modeling error [27], worst-case estimation error bound in presence of bounded noise [28], and $\ell_1 - \ell_2$ estimation scheme [29]. And some iterative resilient observer designs has been proposed for determinant system, such as event-trigger Luenburger Observer [30] and Gramian-based estimator [31]. For stochastic system, several attack-resilient control designs have also been studied [32], [33]. Furthermore, to push the limit on the number of compromised sensors below which secure estimation would be guaranteed, the prior information obtained from detection approaches could be considered in the resilient estimation design. Existing results have used such information as measurement prior, support prior and state prior. In our previous work [34]–[36], resiliency of the observer is enhanced by utilizing a measurement prior, generated by Gaussian process regression, in a constrained ℓ_1 observer scheme. And in [37], a resilient unscented Kalman filter was proposed based on a improved support prior. In [17], a weighted ℓ_1 observer is designed to exploit the support prior more rigorously.

Contribution: This paper proposes an enhanced resilient solution for linear CPSs subject to FDI attacks. It is a hybrid physics-based and data-driven, detection-based resilient estimation scheme. To train the data-driven detection method, abundant adversarial attack examples are often required. Model-based attack generators such as [17], [38] generally adopt optimization-based mechanisms which are computationally expensive, even NP-hard. Most traditional data-driven attack generators [39], [40], in literature, still require the attack examples with labels for training. In this paper, we proposed an unsupervised generative model (GM) trained by multiple physics-based discriminators derived from the plant model. Secondly, due to the inherent uncertainty on the precision of data-driven detection and localization approaches, a pruning algorithm is proposed to improve the localization precision further without training. The resulting precision guarantee is quantified, subject to the aggressiveness of the pruning algorithm. Thus, the states of the system can be recovered correctly, thereby maintaining the operational performance of the CPS while attack is underway.

The reminder of the paper is organized as follows: All the notations and the necessary mathematical tools used in the development are given in Section II. In Section III, the model of CPS is described. In Section IV, a formal definition of successful FDI attack is presented, and a data-driven generative model is proposed to generate successful FDI attacks. In Section V, detection and localization approaches based on multi-class MLP is designed and a pruning algorithm is included to improve the localization precision. In Section VI, a resilient ℓ_2 observer design is done to maintain correct state estimation under adversarial attacks. Conclusion remarks follow in Section VIII.

II. Notation and Preliminary

The following notations and definitions are used throughout the paper: $\mathbb{R}, \mathbb{R}^n, \mathbb{R}^{n\times m}$ denote the space of real numbers, real vectors of length n and real matrices of n rows and m columns respectively. $\mathbb{R}_+$ denotes the space of positive real numbers. Normal-face lower-case letters $(e.g.\ x \in \mathbb{R})$ are used to represent real scalars, bold-face lower-case letters $(e.g.\ \mathbf{x} \in \mathbb{R}^n)$ represent vectors, while normal-face upper-case letters $(e.g.\ X \in \mathbb{R}^{n\times m})$ represent matrices. $X^\top$ denotes the transpose of the matrix X. The spectral radius of the square matrix X is denoted by $\rho(X)$. $X^\dagger \triangleq \left(X^\top X\right)^{-1} X^\top$ denotes the Moore-Penrose inverse. $\mathbf{1}_n$ and I_n denote vector of ones and the identity matrix of size n respectively. Let $\mathcal{T} \subseteq \{1,\dots,n\}$, then, for a matrix $X \in \mathbb{R}^{m\times n}$, $X_\mathcal{T} \in \mathbb{R}^{|\mathcal{T}|\times m}$ is the sub-matrix obtained by extracting the rows of X corresponding to the indices in $\mathcal{T}$. $\mathcal{T}^c$ denotes the complement of a set $\mathcal{T}$ and the universal set on which it is defined will be clear from the context. We use I_T (or simply as I when T is clear from context) to represent a T-time window $[i-T+1, i]$. In the same vein, $I-1$ is used to denote the time window $[i-T, i-1]$ accordingly. The support of a vector $\mathbf{x} \in \mathbb{R}^n$ is a set of the indices of nonzero entries in $\mathbf{x}$, defined as $\mathsf{supp}(\mathbf{x}) \triangleq \{i \subseteq \{1,\dots,n\} | \mathbf{x}_i \neq 0\}$. A vector $\mathbf{x} \in \mathbf{R}^n$ is said to be k-sparse if $|\mathsf{supp}(\mathbf{x})| \leq k$, and Σ_k denotes the subspace of k-sparse vectors. a moving-horizon vector $\mathbf{x}_I \in \Sigma_k$ means all composed vectors $\mathbf{x}_j \in \Sigma_k, j \in I$. $\mathsf{argsort}\downarrow(\mathbf{x})$ denotes a function that returns the sorted indices of vector $\mathbf{x}$ in descending order of the magnitude of $\mathbf{x}_i$. The symbol $\odot$ denotes element-wise multiplication of two vectors and is defined as $\mathbf{z} = \mathbf{x} \odot \mathbf{y}$, where $\mathbf{z}_i = \mathbf{x}_i\mathbf{y}_i$. The operator $\|\mathbf{z}\|_{1,\mathbf{w}} \triangleq \sum_{i=1}^n \mathbf{w}_i\mathbf{z}_i$ is a weighted 1-norm of a vector $\mathbf{z} \in \mathbb{R}^n$ with the weight vector $\mathbf{w} \in \mathbb{R}^n$. A continuous function $f : [0, a) \to [0, \infty)$ is said to belong to class $\mathcal{K}$ if it is strictly increasing and $f(0) = 0$. Both e^x and $\exp(x)$ are used to represent the exponential function. The symbol $*$ denotes the convolution operator of two vectors; given two vectors $\mathbf{u} \in \mathbb{R}^m$ and $\mathbf{v} \in \mathbb{R}^n$,

$$\mathbf{u} * \mathbf{v} \triangleq \sum_j \mathbf{u}_j \mathbf{v}_{k+1-j}.$$

The symbol $\mathbb{E}$ denotes the expectation operator of a random variable. $z \sim \mathcal{B}(1,p)$ represents a Bernoulli distributed random variable z with $\mathbb{E}[z] = p$, then the sum of independent Bernoulli random variables $\{z_1, z_2, \cdots, z_N\}$ with $z_i \sim \mathcal{B}(1,p_i)$ satisfies Poisson-binomial distribution, and the

closed-form expression of the probability density function (pdf) of $\sum_i z_i$ is given by [41]:

$$\mathsf{Pr}\left\{\sum_{i=1}^{N} z_i = k\right\} = \mathbf{r}_k, \quad k = 0, \cdots, N, \tag{1}$$

where

$$\mathbf{r} = \prod_{i=1}^{N} p_i \cdot \begin{bmatrix} -s_1 \\ 1 \end{bmatrix} * \begin{bmatrix} -s_2 \\ 1 \end{bmatrix} * \cdots * \begin{bmatrix} -s_N \\ 1 \end{bmatrix}, \tag{2}$$

with $s_i = -\frac{1-p_i}{p_i}$. The following lemma presents a result that will be useful for later developments in this paper.

Lemma II.1. *Consider a non-decreasing convex function* $\Phi : \mathbb{R} \to \mathbb{R}_+$ *with* $\Phi(0) = 1$*; Then*

$$\mathbb{E}[\Phi(z)] \geq \mathit{Pr}\{z > 0\}$$

.

Proof. Based on the definition of expectation,

$$\begin{aligned}\mathbb{E}[\Phi(z)] &= \sum_{x\in\mathbb{R}} \Phi(x)\mathsf{Pr}\{z = x\} \\ &= \sum_{x\leq 0} \Phi(x)\mathsf{Pr}\{z = x\} + \sum_{x>0} \Phi(x)\mathsf{Pr}\{z = x\}.\end{aligned}$$

Since $\Phi(x) > 0$, then

$$\mathbb{E}[\Phi(z)] \geq \sum_{x>0} \Phi(x)\mathsf{Pr}\{z = x\}.$$

Also, since Φ is non-decreasing and $\Phi(0) = 1$, we have that $\Phi(x) > 1 \ \ \forall x > 0$. Then

$$\mathbb{E}[\Phi(z)] \geq \sum_{x>0} \mathsf{Pr}\{z = x\} = \mathsf{Pr}\{z > 0\}.$$

□

III. Model Development

The following linear model is considered to describe the physical behavior of a CPS:

$$\begin{aligned}\mathbf{x}_{i+1} &= A\mathbf{x}_i + B\mathbf{u}_i \\ \mathbf{y}_i &= C\mathbf{x}_i\end{aligned} \tag{3}$$

where $\mathbf{x}_i \in \mathbb{R}^n$ is the state vector, $\mathbf{u}_i \in \mathbb{R}^p$ the control input and $\mathbf{y}_i \in \mathbb{R}^{m_1}$ the sensor measurement. The corresponding appropriately dimensioned system matrices A, B, C are assumed to satisfy the following.

- A1. The pair (A, B) is controllable
- A2. There exists a positive integer $k_0 \leq m_1$ such that the pair $(A, C_{\mathcal{S}})$ is observable for all $\mathcal{S} \subset \{1, 2, \cdots, m_1\}$ with $|\mathcal{S}| \geq k_0$
- A3. $\rho\left(A^T\right) > 0$, where the integer T is a specified horizon parameter.

To estimate the system states from sensor measurements, the following receding horizon observer is considered:

$$\hat{\mathbf{x}}_i = A^T\mathbf{z} + F\mathbf{u}_{I-1}, \tag{4}$$

where

$$F = \begin{bmatrix} A^{T-1}B & A^{T-2}B & \dots & B \end{bmatrix},$$

and

$$\begin{aligned}\mathbf{z} &= \arg\min_{\mathbf{x}\in\mathbb{R}^n} \|\mathbf{y}_I - H_1\mathbf{u}_{I-1} - H_0\mathbf{x}\|_2 \\ &= H_0^\dagger\mathbf{y}_I - H_0^\dagger H_1\mathbf{u}_{I-1},\end{aligned} \tag{5}$$

where

$$\begin{aligned}H_0 &= \begin{bmatrix} CA \\ CA^2 \\ \vdots \\ CA^T \end{bmatrix} \\ H_1 &= \begin{bmatrix} CB & 0 & \cdots & 0 \\ CAB & CB & \cdots & 0 \\ \vdots & \vdots & \ddots & \vdots \\ CA^{T-1}B & CA^{T-2}B & \cdots & CB \end{bmatrix}.\end{aligned} \tag{6}$$

Thus,

$$\hat{\mathbf{x}}_i = A^T H_0^\dagger \mathbf{y}_I + \left(F - A^T H_0^\dagger H_1\right)\mathbf{u}_{I-1}, \tag{7}$$

Consequently, the following control law is considered:

$$\mathbf{u}_i = K_p\hat{\mathbf{x}}_i, \tag{8}$$

where $\hat{\mathbf{x}}_i \in \mathbb{R}^n$ is the state estimate in (7), which is well known (see [42] for example) to achieve $\lim_{i\to\infty} \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2 = 0$, and the gain matrix K_p is designed to achieve $\rho(A + BK_p) \in (0, \ 1)$. Substituting (8) into the dynamics in (3) yields

$$\begin{aligned}\mathbf{x}_{i+1} &= A\mathbf{x}_i + BK_p\hat{\mathbf{x}}_i \\ &= (A + BK_p)\mathbf{x}_i + BK_p(\hat{\mathbf{x}}_i - \mathbf{x}_i),\end{aligned}$$

which implies that

$$\mathbf{x}_i = (A + BK_p)^i \mathbf{x}_0 + \sum_{j=0}^{i} (A + BK_p)^{i-j} BK_p(\hat{\mathbf{x}}_j - \mathbf{x}_j).$$

Thus,

$$\begin{aligned}\sum_{i=0}^{\infty} \|\mathbf{x}_i\|_2 \leq &\left(\sum_{i=0}^{\infty} \rho\left((A + BK_p)^i\right)\right)\|\mathbf{x}_0\|_2 \\ &+ \sum_{i=0}^{\infty}\sum_{j=0}^{i} \rho\left((A + BK_p)^{i-j} BK_p\right)\|\hat{\mathbf{x}}_j - \mathbf{x}_j\|_2.\end{aligned}$$

Therefore, the resulting closed loop dynamics satisfy

$$\sum_{i=0}^{\infty} \|\mathbf{x}_i\|_2 \leq \alpha_0 + \varphi\left(\sum_{i=0}^{\infty} \|\hat{\mathbf{x}}_i - \mathbf{x}_i\|_2\right)$$

for some positive constant $\alpha_0 < \infty$ and a class $\mathcal{K}$ functional φ. This implies that the origin of the closed loop dynamics is asymptotically stable.

IV. Automated Attack Generation

In this section, a generative model is developed for generating successful FDI attacks targeting the model in (3). Including attack signals in the CPS model in (3) yields the adversarial measurement model

$$\mathbf{y}_I = H_0\mathbf{x}_{i-T} + H_1\mathbf{u}_{I-1} + \mathbf{e}_I, \tag{9}$$

where $\mathbf{e}_{1:T} = \begin{bmatrix} \mathbf{e}_{i-T+1}^\top & \mathbf{e}_{i-T+2}^\top & \dots & \mathbf{e}_i^\top \end{bmatrix}^\top$ contains the attack vectors within the moving window $[i-T+1,\ i]$. Consequently, substituting (9) into (7) yields

$$\begin{aligned}\hat{\mathbf{x}}_i &= A^T\mathbf{x}_{i-T} + F\mathbf{u}_{I-1} + A^T H_0^\dagger \mathbf{e}_I \\ &= \mathbf{x}_i + A^T H_0^\dagger \mathbf{e}_I. \end{aligned} \tag{10}$$

Thus, the instantaneous estimation error is given by

$$\begin{aligned}\tilde{\mathbf{x}}_i &= \hat{\mathbf{x}}_i - \mathbf{x}_i \\ &= A^T H_0^\dagger \mathbf{e}_I. \end{aligned} \tag{11}$$

In general, attackers aim to achieve maximum perturbation of certain critical system output without triggering an alarm from the bad data detection mechanism. For this paper, a residual-based bad data detection scheme is used. Let

$$\mathbf{r}_I \triangleq \mathbf{y}_I - H_1\mathbf{u}_{I-1} - H_0\mathbf{z}, \tag{12}$$

where $\mathbf{z}$ is given in (5), be the residual vector associated with the receding horizon estimator in (7). Given a threshold value $\tau \in \mathbb{R}$, the bad data indicator for the windowed measurement history is given by

$$\mathsf{BDD}(\mathbf{y}_I) = \begin{cases} 0 & \text{if } \|\mathbf{r}_i\| \le \tau \\ 1 & \text{otherwise.} \end{cases} \tag{13}$$

Substituting (5) and (9) into (12) yields

$$\begin{aligned}\mathbf{r}_I &= \left(I - H_0H_0^\dagger\right)\left(\mathbf{y}_I - H_1\mathbf{u}_{I-1}\right) \\ &= \left(I - H_0H_0^\dagger\right)\left(H_0\mathbf{x}_{i-T} + \mathbf{e}_I\right). \end{aligned} \tag{14}$$

Thus, from the attacker's perspective, it is desirable to keep the quantity $\left\|\left(I - H_0H_0^\dagger\right)\mathbf{e}_I\right\|_2$ as small as possible.

Next, let $\mathbf{y}_c = C_c\mathbf{x}$ be a vector of critical output targeted by the attacker. We remark that this is not an attempt to identify a priori the particular outputs targeted by an attacker. Rather, we postulate that there are critical functional outputs which the resilient system designer desires to protect against adversarial targeting. Thus, we define FDI attacks which target those measurements specifically so that the resulting resilient observer can offer the desired protection. Substituting the control law in (8) into the linear model in (3) yields the closed loop system

$$\begin{aligned}\mathbf{x}_{i+1} &= \bar{A}\mathbf{x}_i + \bar{B}\tilde{\mathbf{x}}_i, \\ &= \bar{A}\mathbf{x}_i + \bar{B}A^T H_0^\dagger \mathbf{e}_I. \end{aligned} \tag{15}$$

where $\bar{A} = A + BK_p$ and $\bar{B} = BK_p$. Therefore,

$$\mathbf{y}_{c_{i+1}} = C_c\bar{A}\mathbf{x}_i + C_c\bar{B}A^T H_0^\dagger \mathbf{e}_I. \tag{16}$$

Thus, from the attacker perspective, it is desirable to keep the quantity $\left\|C_c\bar{B}A^T H_0^\dagger \mathbf{e}_I\right\|_2$ as big as possible.

We now have all the ingredients to formally define and synthesize the generative model for successful FDI attacks in CPSs.

Definition 1 (Successful FDI attack [16])**.** *Consider the CPS in* (3)*. Given a positive integer $k < m$, the attack sequence $\mathbf{e}_I \in \Sigma_k \subset \mathbb{R}^{(T+1)m}$, is said to be (ϵ, α)-successful if*

$$\left\|C_c\bar{B}A^T H_0^\dagger \mathbf{e}\right\|_2 \ge \alpha \text{ and } \left\|\left(I - H_0H_0^\dagger\right)\mathbf{e}\right\|_2 \le \epsilon. \tag{17}$$

In the above definition, k quantifies the attack sparsity level per time. Specifically, it is the maximum number of attacks allowed at each time index. Examples of such successful FDI attacks can be found in literature; a physics-based FDI attack generator capable of fooling ℓ_2 observer, such as Kalman filer, can be found in [16] and an optimization-based FDI attack generator capable of fooling stronger ℓ_1 observer was proposed in [17]. However, these FDI attacks are either computationally too expensive or too conservative to implement on fast real-time systems. In order to circumvent these limitations, we drew inspiration from the well-know generative adversarial networks (GAN) [43] to build generator models for the elements of the set specified by inequalities in (17).

The Definition 1 defines a target set of successful attack signals of the form:

$$\mathcal{S}(\alpha, \epsilon) \triangleq \left\{\mathbf{e} \mid \|M_1\mathbf{e}\|_2 \ge \alpha, \|M_2\mathbf{e}\|_2 \le \epsilon\right\}. \tag{18}$$

Thus, the goal of generative model is to learn how to generate elements of that set.

Definition 2. *Given a set $\mathcal{S} \subset \mathbb{R}^n$ and a prior distribution $P_z(\mathbf{z})$, a generative model is a mapping of the form $f : \mathbf{z} \mapsto \mathbb{R}^n$ such that $f(\mathbf{z}) \in \mathcal{S}$ with a high probability.*

We consider a generative model of the form $G(\mathbf{z}; \boldsymbol{\theta}_g)$, where $\boldsymbol{\theta}_g$ is a tunable parameter vector that will be prescribed to maximize the likelihood of generating elements in $\mathcal{S}$. Specifically, G is taken as an artificial neural network whose input is a random number $\mathbf{z}$ sampled from a prior distribution P_z, with weights and biases contained in $\boldsymbol{\theta}_g$. Next, similarly to the GAN framework, $G(\mathbf{z}; \boldsymbol{\theta}_g)$ is trained using a fixed discriminator defined by the boundary functions $\|M_1\mathbf{e}\|_2$ and $\|M_2\mathbf{e}\|_2$ of $\mathcal{S}$.

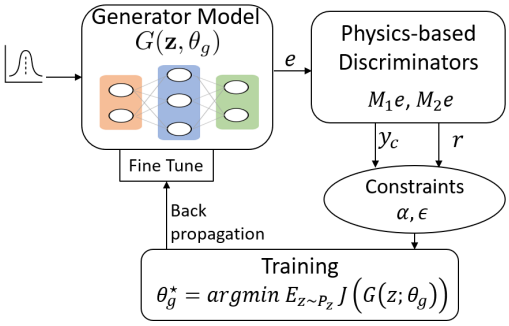

Fig. 1. Physics-based generative model and training process

The training process is shown in Fig. 1. The output of the generator is passed, through the discriminator boundary functions, to a loss function designed such that the weights of the generator network are trained to ensure that the outputs of the discriminators satisfy the constraints. Since the constraints are threshold-based, we choose an indicator-like loss function:

$$J(\mathbf{e}) = \mathsf{ReLU}(\alpha - \|M_1\mathbf{e}\|_2) + \mathsf{ReLU}(\|M_2\mathbf{e}\|_2 - \epsilon). \tag{19}$$

Consequently, the generator is trained by solving the unconstrained optimization problem

$$\boldsymbol{\theta}_g^* = \arg\min \mathop{\mathbb{E}}_{\mathbf{z}\sim P_z} J(G(\mathbf{z}; \boldsymbol{\theta}_g)) \tag{20}$$

via back propagation.

V. Detection and Localization

Through an automated attack generation by a well-trained generative model, an adversarial training dataset could be constructed and used to improve the precision of the resulting attack detection and localization algorithm. In this section, a detection and localization approach is given by training a multi-layer perceptron (MLP) on the generated attack dataset. In addition, by analyzing the uncertainty on the MLP's output, a pruning algorithm is proposed to improve the localization precision further.

The term `attack location` refers to sensor channels (or indices) through which the attack signals are added to the system. Let $\mathcal{T}$ denote the actual attack support, then we define its indicator element-wise as:

$$\mathbf{q}(i) = \begin{cases} 0 & \text{if } i \in \mathcal{T} \\ 1 & \text{otherwise.} \end{cases} \tag{21}$$

Next, the definitions of detection and localization are given:

Definition 3 (Detection)**.** *Given a measurement vector* $\mathbf{y} \in \mathbb{R}^m$*, detection is a function* $D : \mathbb{R}^m \to \{0, 1\}$ *whose output* $D(\mathbf{y})$ *indicates the existence of attack signal in the input* $\mathbf{y}$*.*

Definition 4 (Localization)**.** *Given a measurement vector* $\mathbf{y} \in \mathbb{R}^m$*, localization is a function* $L : \mathbb{R}^m \to \{0, 1\}^m$ *whose output indicates detection result for each measurement channel. In particular, it is an estimate of the attack support* $\mathcal{T}$*, denoted by* $\hat{\mathcal{T}}$*, and its indicator* $\hat{\mathbf{q}}$ *is defined similarly to* (21)*.*

As a universal function approximators, shown in Cybenko's theorem [44], MLPs are used to describe mathematical models via a regression scheme. Since classification is a special regression process with categorical response, MLPs are often designed as classification algorithms. Attack detection and localization is a multi-line binary classification task. With adequate training dataset generated by the automated attack generator, MLP's advantages of being a learning network could be fully utilized.

The MLP used in this paper is a three-layer neural network containing two *LeakyReLU* hidden layers and one *Sigmoid* output layer, and the final classification outputs are 0 (for attack) if the corresponding MLP output is less than 0.5, otherwise they are 1 (for no attack). The output layer's size is the same as the input size and equal to the number of attacks. Consequently, the MLP is given by

$$g(\mathbf{y}; \boldsymbol{\theta}) \triangleq \phi\left(\sum_{i=1}^{n} w_i \mathbf{y}_i + b\right), \tag{22}$$

where ϕ is the activation function of the output layer and $\boldsymbol{\theta}$ is a vector of tunable weights and biases. Then, the localization output is given by

$$\hat{\mathbf{q}}_i = \begin{cases} 1 & \text{if } g(\mathbf{y}_i; \boldsymbol{\theta}) \geq 0.5 \\ 0 & \text{otherwise.} \end{cases} \tag{23}$$

Next, a supervised learning algorithm, via back propagation, is used to train the MLP based on the below binary cross-entropy loss function (log-loss function):

$$J(\boldsymbol{\theta}) = \frac{1}{N}\sum_{i=1}^{N}\Big(\mathbf{q}^{(i)} \log g(\mathbf{y}^{(i)}; \boldsymbol{\theta}) + (\mathbf{1} - \mathbf{q}^{(i)}; \boldsymbol{\theta}) \log(\mathbf{1} - g(\mathbf{y}^{(i)}; \boldsymbol{\theta}))\Big) \tag{24}$$

where $(\mathbf{y}^{(i)}, \mathbf{q}^{(i)})$ is the ith training data, and N is the total number of data points.

Since the MLP-based attack detection and localization method described above is a multi-line binary classification, the agreement between the localization result $\hat{\mathbf{q}}$ and the actual attack support $\mathbf{q}$ can be described by a Bernoulli distributed agreement variable $\epsilon_i \sim \mathcal{B}(1, \mathbf{p}_i)$ as

$$\mathbf{q}_i = \epsilon_i \hat{\mathbf{q}}_i + (1 - \epsilon_i)(1 - \hat{\mathbf{q}}_i), \tag{25}$$

where $\mathbf{p}_i = \mathbb{E}[\epsilon_i] = \mathsf{Pr}\{\epsilon_i = 1\}$ is the confidence of the localization algorithm at i channel, which is often obtained by the true rate in *Receiver Operator Characteristic* (ROC) of the algorithm.

Consequently, the precision of the resulting support prior indicator $\hat{\mathbf{q}}$ is defined as follows:

Definition 5 (Positive Prediction Value, Precision, PPV [45])**.** *Given an estimate* $\hat{\mathbf{q}} \in \{0, 1\}^N$ *of an unknown attack support indicator* $\mathbf{q} \in \{0, 1\}^N$*, PPV is the proportion of* $\mathbf{q}$ *that is correctly identified in* $\hat{\mathbf{q}}$*. It is given by*

$$PPV = \frac{\|\mathbf{q} \odot \hat{\mathbf{q}}\|_{\ell_0}}{\|\hat{\mathbf{q}}\|_{\ell_0}}. \tag{26}$$

Next, with the attack localization result $\hat{\mathcal{T}}$, the estimated support of safe nodes can be represented as $\hat{\mathcal{T}}^c = \{j_1, j_2, \cdots, j_{|\hat{\mathcal{T}}^c|}\}$. A *Pruning Algorithm* is proposed to prune the most possible erroneous localization results in $\hat{\mathcal{T}}^c$.
The first step is to obtain a trust integer $l_\eta (\leq Tm)$ based on the confidence of the localization results $\mathbf{p}$. Given a reliability level $\eta \in (0, 1)$, the trust integer l_η is defined as the maximum number of safe nodes correctly localized in $\hat{\mathcal{T}}^c$ with a probability of at least η,

$$l_\eta \triangleq \mathsf{max}\left\{k \middle| \mathsf{Pr}\left\{\sum_{i \in \hat{\mathcal{T}}^c} \epsilon_i \geq k\right\} \geq \eta\right\}. \tag{27}$$

Let

$$\bar{\mathbf{r}} \triangleq \begin{bmatrix} r_0 \\ r_1 \\ \vdots \\ r_{|\hat{\mathcal{T}}^c|} \end{bmatrix} = \prod_{i=1}^{|\hat{\mathcal{T}}^c|} \mathbf{p}(j_i) \begin{bmatrix} -s_{j_1} \\ 1 \end{bmatrix} * \begin{bmatrix} -s_{j_2} \\ 1 \end{bmatrix} * \cdots * \begin{bmatrix} -s_{j_{|\hat{\mathcal{T}}^c|}} \\ 1 \end{bmatrix}, \tag{28}$$

where $s_{j_i} = -\frac{1-\mathbf{p}_i}{\mathbf{p}_i}$. Then, $\mathsf{Pr}\{\sum_{i \in \hat{\mathcal{T}}^c} \epsilon_i = k\} = \bar{\mathbf{r}}_k$. Thus (27) becomes

$$l_\eta = \mathsf{max}\left\{k \middle| \sum_{i=0}^{k} \bar{\mathbf{r}}_i \leq 1 - \eta\right\}. \tag{29}$$

The second step is to obtain a *Pruned Support Prior* $\hat{\mathcal{T}}_\eta^c$ of size l_η through a robust extraction:

$$\hat{\mathcal{T}}_\eta^c = \left\{\mathsf{argsort}\downarrow (\mathbf{p} \odot g(\mathbf{y};\boldsymbol{\theta}))\right\}_1^{l_\eta}, \tag{30}$$

where $\{\cdot\}_1^{l_\eta}$ is an index extraction from the first elements to l_η elements. To obtain a pruned support prior consistent with the model assumptions in Section III, its size should satisfy $l_\eta \geq k_0$, where k_0 is a lower bound of $\hat{\mathcal{T}}_\eta^c$ such that $(A, C_{\hat{\mathcal{T}}_\eta^c})$ is observable. This can be guaranteed by the proper choice of the reliability level η, as given in the following theorem.

Theorem V.1. *Given a measurement redundancy preservation parameter $k_0 > 0$, the size of pruned support prior satisfies $l_\eta \geq k_0$ if*

$$\eta \leq 1 - \sum_{i=0}^{|\hat{\mathcal{T}}^c|} e^{k_0 - i} \bar{\mathbf{r}}_i, \tag{31}$$

where $\bar{\mathbf{r}}$ is given by (28)*, and $\hat{\mathcal{T}}^c$ is the estimated support of safe nodes from the attack localization algorithm.*

Proof. According to (27), a sufficient condition for $l_\eta \geq k_0$ is

$$\mathsf{Pr}\left\{\sum_{i\in\hat{\mathcal{T}}^c} \epsilon_i \geq k_0\right\} \geq \eta,$$

which is equivalent to:

$$\mathsf{Pr}\left\{k_0 - \sum_{i\in\hat{\mathcal{T}}^c} \epsilon_i \leq 0\right\} \geq \eta$$

and

$$\mathsf{Pr}\left\{k_0 - \sum_{i\in\hat{\mathcal{T}}^c} \epsilon_i \geq 0\right\} \leq 1 - \eta. \tag{32}$$

According to Lemma II.1, let $z = k_0 - \sum_{i\in\hat{\mathcal{T}}^c} \epsilon_i$, and take $\Phi(z) = \exp(z)$ yields a sufficient condition for (32) as

$$\begin{aligned} &\mathbb{E}\left[\exp\left(k_0 - \sum_{i\in\hat{\mathcal{T}}^c} \epsilon_i\right)\right] \leq 1 - \eta, \\ \equiv\ & e^{k_0}\mathbb{E}\left[\exp\left(-\sum_{i\in\hat{\mathcal{T}}^c} \epsilon_i\right)\right] \leq 1 - \eta. \end{aligned} \tag{33}$$

Since $\mathsf{Pr}\left\{\sum_{i\in\hat{\mathcal{T}}^c} \epsilon_i = k\right\} = \bar{\mathbf{r}}_k$, it follows that

$$\begin{aligned} \mathbb{E}\left[\exp\left(-\sum_{i\in\hat{\mathcal{T}}^c} \epsilon_i\right)\right] &= \sum_{k=0}^{|\hat{\mathcal{T}}^c|} e^{-k}\mathsf{Pr}\left\{\sum_{i\in\hat{\mathcal{T}}^c} \epsilon_i = k\right\} \\ &= \sum_{k=0}^{|\hat{\mathcal{T}}^c|} e^{-k}\bar{\mathbf{r}}_k. \end{aligned} \tag{34}$$

Combining (33) and (34) yields

$$\sum_{i=0}^{|\hat{\mathcal{T}}^c|} e^{k_0 - i}\bar{\mathbf{r}}_i \leq 1 - \eta. \tag{35}$$

Finally, an upper bound on η is obtained as (31). □

Remark 1. *From* (31)*, it is clear that, necessarily, the upper bound should be strictly positive. This yields the following pruning requirement on the localization performance parameters:*

$$\sum_{i=0}^{|\hat{\mathcal{T}}^c|} e^{-i}\bar{\mathbf{r}}_i < e^{-k_0}. \tag{36}$$

If this holds, according to the assumption A2, with at least k_0 safe nodes remaining in the pruned support prior $\hat{\mathcal{T}}_\eta^c$, $(A, C_{\hat{\mathcal{T}}_\eta^c})$ will be observable. Thus the asymptotic convergence of the estimation error is guaranteed.

Moreover, it was shown in [17] that the precision PPV_η of the pruned support prior $\hat{\mathcal{T}}_\eta^c$ given by (30) achieves

$$\mathsf{Pr}\left\{\mathsf{PPV}_\eta = 1\right\} \geq \eta. \tag{37}$$

This indicates that, with the probability of at least η, the measurement nodes contained in $\hat{\mathcal{T}}_\eta^c$ are all safe, and $(A, C_{\hat{\mathcal{T}}_\eta^c})$ is observable according to Remark 1. Thus, with the probability of at least η, the exact state estimation can be achieved by using ℓ_2 observer designed based on $(A, C_{\hat{\mathcal{T}}_\eta^c})$. Finally, the Algorithm 1 summarizes the attack detection and localization using automated attack generation and pruning operation.

Algorithm 1: Attack detection and localization with automated attack generation and pruning

I. **Offline: Dataset generation and training.**
1) Given time horizon T, prepare M_1, M_2 matrices for physics-based discriminator ← (14), (16);
2) Given the number of attacks $|\mathcal{T}|$, randomize attack support $\mathcal{T}$, and obtain its indicator $\mathbf{q}$ ← (21);
3) Given attack indexes α, ϵ, train the generative model (GM) by solving the optimization problem in (20);
4) Generate attack dataset $\{Y\}$ using the trained GM. Evaluate generated attack dataset via simulation.
5) Return the effective subset of attack dataset as the training dataset $\{Y, \mathbf{q}\}$.
6) Train the MLP in (22) using the generated dataset $\{Y, \mathbf{q}\}$ via backpropagation.

II. **Online: Localization and pruning**
1) Input real-time measurements $\mathbf{y}$, obtain the classification result $\hat{\mathbf{q}}$ ← (22), (23);
2) Pruning to obtain the pruned support prior $\hat{\mathcal{T}}_\eta^c(\hat{\mathbf{q}}_\eta)$ ← (29), (30).

Output: $\hat{\mathcal{T}}_\eta^c$ (pruned support prior)

VI. Resilient Estimator

In this section, the ℓ_2 observer in (7) is redesigned based on the pruned support generated by the Algorithm 1 in order to maintain secure state estimation subject to FDI attack. The scheme of the resilient ℓ_2 observer is shown in Fig. 2. The attack detection and localization algorithm identifies the locations of attacks and return an estimate of the support of safe nodes $\hat{\mathcal{T}}^c$. Then the pruning algorithm is used to downselect to a subset of $\hat{\mathcal{T}}^c$ based on the localization confidence. Finally,

the downselected subset of measurements $\mathbf{y}_{\hat{\mathcal{T}}_\eta^c}$ is used in the ℓ_2 observer.

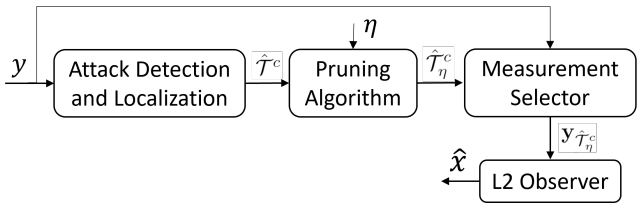

Fig. 2. Schematic diagram of resilient ℓ_2 observer

Given the pruned support $\hat{\mathcal{T}}_\eta^c$ within the moving window $[i-T+1, i]$, the resilient ℓ_2 observer is given by:

$$\hat{\mathbf{x}}_i^{'} = A^T H_{0,\hat{\mathcal{T}}_\eta^c}^{\dagger} \mathbf{y}_{\hat{\mathcal{T}}_\eta^c} + \left(F - A^T H_{0,\hat{\mathcal{T}}_\eta^c}^{\dagger} H_{1,\hat{\mathcal{T}}_\eta^c}\right) \mathbf{u}_{I-1}, \quad (38)$$

where $H_{0,\hat{\mathcal{T}}_\eta^c}, H_{1,\hat{\mathcal{T}}_\eta^c}$ are defined similarly to (6) using $C_{\hat{\mathcal{T}}_\eta^c}$ in place of C, and $\mathbf{y}_{\hat{\mathcal{T}}_\eta^c}$ is part of the measurements indexed in $\hat{\mathcal{T}}_\eta^c$. According to (37), $\mathbf{y}_{\hat{\mathcal{T}}_\eta^c}$ contains only clean measurements with a probability of at least η. The next result gives the error bound of the resulting resilient estimator.

Theorem VI.1. *Given a pruned support $\hat{\mathcal{T}}_\eta^c$ from* (30)*. Suppose η satisfies* (31) *with the underlying attack localization algorithm satisfying the performance criterion in* (36)*. Then, with the probability of at least η, the resilient estimator in* (38) *satisfies the error bound,*

$$\|\hat{\mathbf{x}}_i^{'} - \mathbf{x}_i^{\star}\|_2 \leq \|\hat{\mathbf{x}}_i - \mathbf{x}_i^{\star}\|_2 \quad (39)$$

where $\hat{\mathbf{x}}_i^{'}, \hat{\mathbf{x}}_i$ are the state estimates from (38) *and* (7) *respectively, $\mathbf{x}^\star$ is the actual state vector.*

Proof. Using triangle inequality of vector norm yields

$$\|\hat{\mathbf{x}}_i^{'} - \mathbf{x}_i^{\star}\|_2 = \|\hat{\mathbf{x}}_i^{'} - \hat{\mathbf{x}}_i + \hat{\mathbf{x}}_i - \mathbf{x}_i^{\star}\|_2 \leq \|\hat{\mathbf{x}}_i^{'} - \hat{\mathbf{x}}_i\|_2 + \|\hat{\mathbf{x}}_i - \mathbf{x}_i^{\star}\|_2 \quad (40)$$

Claim: $\|\hat{\mathbf{x}}_i^{'} - \hat{\mathbf{x}}_i\|_2 = 0$

Proof of Claim: Since $\rho(A^\top) > 0$, using the relationship $\hat{\mathbf{x}}_i = A^T\mathbf{z} + F\mathbf{u}_{I-1}$ and $\hat{\mathbf{x}}_i^{'} = A^T\mathbf{z}^{'} + F\mathbf{u}_{I-1}$, it is seen that $\|\hat{\mathbf{x}}_i^{'} - \hat{\mathbf{x}}_i\|_2 = 0$ is equivalent to $\|\mathbf{z}^{'} - \mathbf{z}\|_2 = 0$, where $\mathbf{z}$ and $\mathbf{z}^{'}$ are given by

$$\begin{aligned} \mathbf{z} &= H_0^{\dagger}\mathbf{y}_I - H_0^{\dagger}H_1\mathbf{u}_{I-1} \\ &= H_0^{\dagger}H_0\mathbf{x}_{i-T} \end{aligned} \quad (41)$$

and

$$\begin{aligned} \mathbf{z}^{'} &= H_{0,\hat{\mathcal{T}}_\eta^c}^{\dagger}\mathbf{y}_{\hat{\mathcal{T}}_\eta^c} - H_{0,\hat{\mathcal{T}}_\eta^c}^{\dagger}H_{1,\hat{\mathcal{T}}_\eta^c}\mathbf{u}_{I-1} \\ &= H_{0,\hat{\mathcal{T}}_\eta^c}^{\dagger}H_{0,\hat{\mathcal{T}}_\eta^c}\mathbf{x}_{i-T} + H_{0,\hat{\mathcal{T}}_\eta^c}^{\dagger}\mathbf{e}_{\hat{\mathcal{T}}_\eta^c} \end{aligned} \quad (42)$$

respectively. Since the pruned support prior is generated by (30), then (37) holds, which implies that, with the probability of at least η, $\mathbf{e}_{\hat{\mathcal{T}}_\eta^c} = \mathbf{0}$. Thus, (42) becomes

$$\mathbf{z}^{'} = H_{0,\hat{\mathcal{T}}_\eta^c}^{\dagger}H_{0,\hat{\mathcal{T}}_\eta^c}\mathbf{x}_{i-T}. \quad (43)$$

Then combining (41) and (43) yields

$$\|\mathbf{z}^{'} - \mathbf{z}\|_2 = \left\|(H_{0,\hat{\mathcal{T}}_\eta^c}^{\dagger}H_{0,\hat{\mathcal{T}}_\eta^c} - H_0^{\dagger}H_0)\mathbf{x}_{i-T}\right\|_2. \quad (44)$$

Since η satisfies (31) and the underlying attack localization algorithm satisfies the performance criterion in (36), then $|\hat{\mathcal{T}}_\eta^c| > k_0$ which implies $(A, C_{\hat{\mathcal{T}}_\eta^c})$ is observable according to the assumption A2. Then, based on the definition of $H_{0,\hat{\mathcal{T}}_\eta^c}$ and H_0, they are both full rank. Therefore, in case of $k_0 < m_1$, the Moore-Penrose pseudo-inverses can be directly calculated as $H_{0,\hat{\mathcal{T}}_\eta^c}^{\dagger} = (H_{0,\hat{\mathcal{T}}_\eta^c}^{\top}H_{0,\hat{\mathcal{T}}_\eta^c})^{-1}H_{0,\hat{\mathcal{T}}_\eta^c}^{\top}$ and $H_0^{\dagger} = (H_0^{\top}H_0)^{-1}H_0^{\top}$. Then $\|\mathbf{z}^{'} - \mathbf{z}\|_2 = \|(I-I)\mathbf{x}_{i-T}\|_2 = 0$. The claim is proved. Thus, (40) implies (39). □

By using ℓ_2 observer in (7), the attack-free state estimation error $\|\hat{\mathbf{x}}_i - \mathbf{x}_i^{\star}\|_2$ converges to zero as i goes infinity. According to (39), $\|\hat{\mathbf{x}}_i - \mathbf{x}_i^{\star}\|_2$ also goes to zero in the limit.

VII. Simulation

In this section, the corresponding improvement of resiliency due to the automated attack generation is shown in simulation by implementing Algorithm 1 and the resilient observer in (38) on a water distribution system.

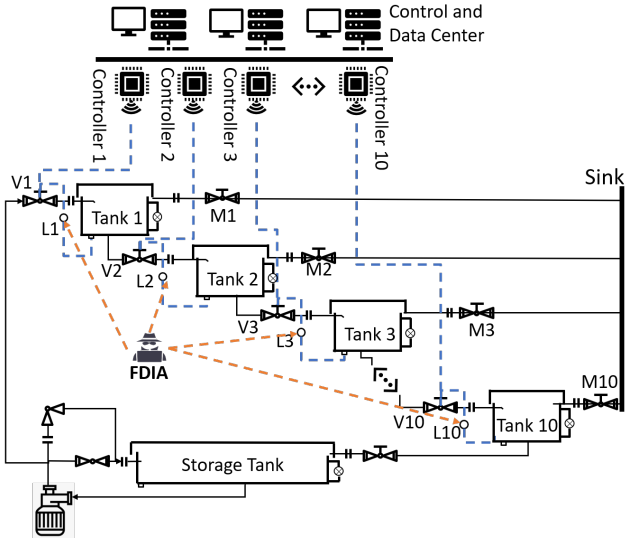

Fig. 3. Block diagram depiction of a water distribution system under FDI attacks (black solid are water pipelines, blue dotted lines are wireless data transmission lines for sensors data and control commands, orange dotted lines are attack injection route)

The water distribution system, shown in Fig. 3, contains 10 distributed operating tanks and 1 storage tank. The goal is to regulate all operating tanks' water levels around desired values. The magnetic valves $V_1 - V_{10}$ at the entrance of operating tanks are controlled to adjust the water level of water tanks, and the manual valves $M_1 - M_{10}$ can be manipulated, at the demand point, to use the water from the corresponding tanks. The effect of these valves is modelled as the random demands $\mathbf{d}$ in (45). The valves at the storage tank are fixed at a constant opening value. For each operating tanks, two sensors, water level meter and pressure sensors (L_1 L_{10}), are installed. The pressure sensors measure the difference of water level between adjoining tanks. The water level adjustment process is approximated by the LTI model:

$$\begin{aligned} \mathbf{x}_{i+1} &= A\mathbf{x}_i + B\mathbf{u}_i - \mathbf{d}_i, \\ \mathbf{y}_i &= C\mathbf{x}_i + \mathbf{v}_i + \mathbf{e}_i. \end{aligned} \quad (45)$$

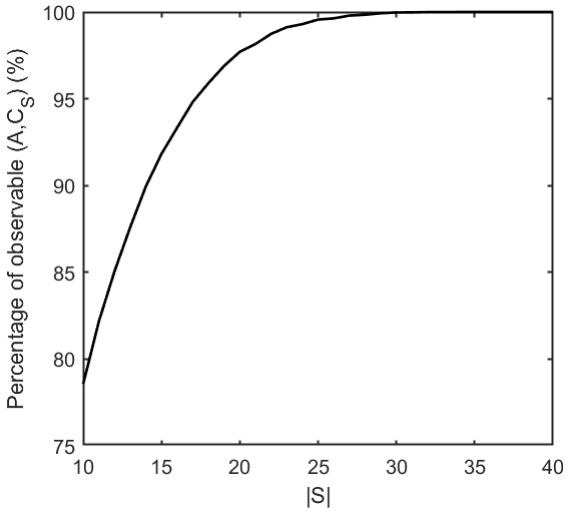

Fig. 4. The percentage of observable $(A, C_{\mathcal{S}})$ increases with the size of the pruning set $|\mathcal{S}|$. The percentage of observable $(A, C_{\mathcal{S}})$ is calculated as the ratio of the number of observable $(A, C_{\mathcal{S}})$ to the total number of samples (=50000).

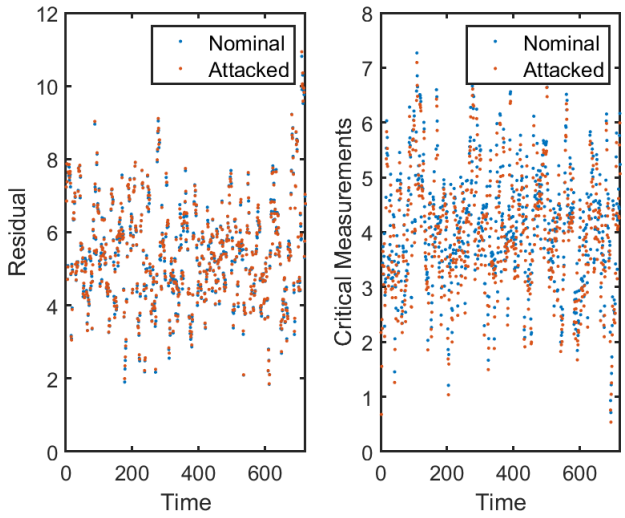

Fig. 5. A testing result of generated attacks injected through 25% measurements nodes of the water distribution system. (attack support $\mathcal{T} = \{2\ 6\ 8\ 11\ 12\ 15\ 16\ 24\ 25\ 33\}$. Left: T-horizon cumulative residual, Right: critical measurement at each time step.)

where, $\mathbf{x}, \mathbf{u} \in \mathbb{R}^{10}$, $\mathbf{y}, \mathbf{v} \in \mathbf{R}^{40}$ are vector of water level, control input of magnetic valves, sensor measurements, and measurement noise respectively, $\mathbf{d} \in \mathbb{R}^{10}$ is the vector of random demands, and $\mathbf{e} \in \mathbb{R}^{40}$ is the injected FDI attacks. Given target water level $\mathbf{x}_d = 0.5\mathbf{1}_{10}$, a proportional controller is used

$$\mathbf{u}_i = -K(\mathbf{x} - \mathbf{x}_d) - B^{-1}(A - I_{10})\mathbf{x}_d.$$

The critical measurement in this scenario is the sum of all water levels of operating tanks, thus, the critical measurement matrix is given by $C_c = [1\ \ 1\ \ \cdots\ \ 1] \in \mathbb{R}^{1\times 10}$. Fig. 4 shows the probability of the pair (A, C) remaining observable after pruning with the random support $\mathcal{S}$ of different size. The result indicates that the system is still observable for certain if at least 27 measurements remain after pruning. According to the Remark 1, the asymptotic convergence of estimation error is guaranteed after the pruning algorithm if the MLP localization algorithm satisfies the condition in (36) with $k_0 = 27$. If this condition is violated, the pruning algorithm shall not be used.

Next, the Algorithm 1 is implemented and the results of simulation are shown in three steps: automated attack generation, training data preparation and MLP training, and Monte-Carlo simulation for resilient ℓ_2 observer with different attack detection and localization strategies.

A. Automated attack generation

A fixed time step of $T_s = 0.01s$ is used and the moving-window size is set as $T = 8T_s$. With different number of attacks, the transfer matrices M_1, M_2 are calculated based on the corresponding random attack support, then the generator (20) is trained as shown in Fig. 1. Due to the uncertainty in the discriminator caused by measurement noise and random demands, the threshold values are chosen from experience as $\alpha = 0.2, \epsilon = 0.6$. The trained generator is then used to generate attacks during simulation and also augment the dataset used for training the localization MLP.

Fig. 5 shows a typical example of the performance of 10 attacks. The left figure shows that the T-horizon cumulative residual under attacks has small deviation from the nominal T-horizon cumulative residual, which means the attacks are hiding in the noise. However, the right figure shows that the critical measurement has obvious deviation from nominal critical measurement after attacks are injected, which means the generated attacks inject targeted biases in the system successfully.

B. MLP training

To show one of the merits of the proposed automated attack generation scheme, two training datasets are prepared: one produced by random additive signals on measurements and the other augments the dataset with attack samples generated by the trained generator. Next, two MLP classifiers are trained on the two datasets respectively. Fig. 6 shows that the MLP classifier (MLP2) trained by the second dataset has better localization precision. The mean precision of MLP1 is 73.87%, and the mean precision of MLP2 is 83.31% on a testing set of successful FDI attacks. Clearly, MLP2 has successfully shifted the localization precision to the right, indicating better performance overall.

As shown in Fig. 4, full observability can be guaranteed with certainty if $k_0 = 27$ measurements remains after pruning. For this, we evaluate the sufficient condition in (36) on MLP1 and MLP2. First, the two MLPs are run on 200000 cases generated by the trained generative model. From the results, the confidence vectors of the MLPs are estimated by calculating the true rates at all 40 measurement nodes; for node i, $\mathbf{p}_i = \frac{TP_i + TN_i}{P_i + N_i}$. Next, a sample based experiment is performed. For each support size $|\hat{\mathcal{T}}_j^c| \in \{25, 26, \ldots, 40\}$, we generated $\min\left\{{}^{40}C_{|\hat{\mathcal{T}}_j^c|}, 50000\right\}$ unique supports $\hat{\mathcal{T}}^c$, and

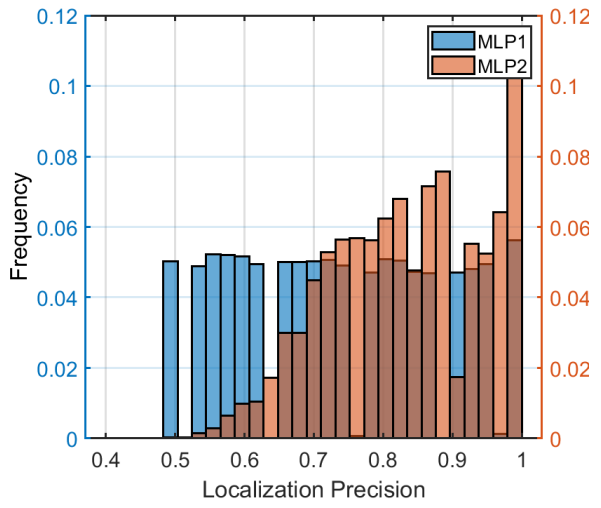

Fig. 6. A sample-based comparison of the performance of two MLP classifiers on a testing set of successful FDI attacks. (MLP1: trained only with random attack generation, MLP2: trained with random attack generation and the proposed automated attack generation, 50000 test cases are used.)

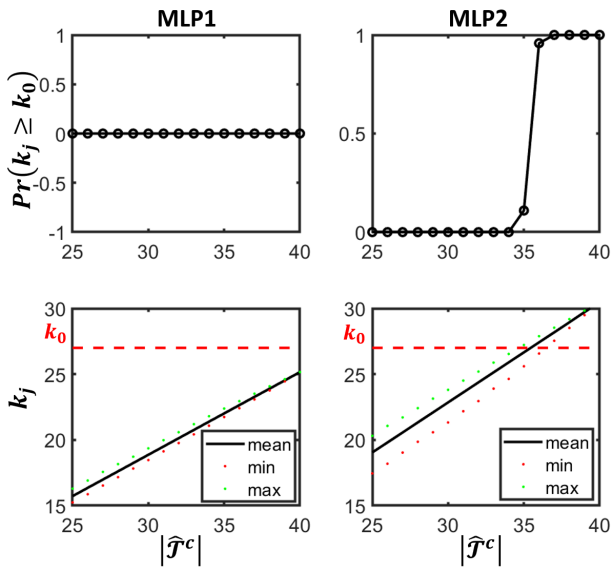

Fig. 7. A sample-based comparison of post-pruning observability for MLP1 and MLP2. (50000 test cases are used. The y axis for the top two subplots $Pr(k_j \geq k_0)$ is the sample estimate of the probability of at least k_0 measurements remaining after pruning. The black lines in the bottom subplots show the mean value of k_j that is the number of measurements remaining after pruning, while the red and green dots are the minimum and maximum values of k_j's respectively.

evaluated the right hand side of (36) for each unique support as follows:

$$k_j = -\ln\left(\sum_{i=0}^{|\hat{\mathcal{T}}_j^c|} e^{-i}\bar{\mathbf{r}}_i\right),$$

where the Poisson-binomial probability density vector $\bar{\mathbf{r}}_i$ is calculated, using the confidence vectors of MLPs, based on (28). For each $|\hat{\mathcal{T}}_j^c| \in \{25, 26, \ldots, 40\}$, and each MLP, the sample probability of the event $k_j \geq k_0$ and the expected value of k_j are shown in Fig. 7. The results show that MLP2 has a bigger chance of preserving post-pruning observability. Indeed, according to the figure, post-pruning observability cannot be guaranteed when MLP1 is used. By considering the sample-based analysis in Fig. 6 and Fig. 7 together, it is seen that MLP performs better with the proposed automated attack generation.

Next we shall present a comparison of MLP1 and MLP2 in the resilient ℓ_2 estimation scheme for the water distribution system to show their effects on the estimator's resiliency.

C. Resilient ℓ_2 observer

Finally, the trained MLPs with pruning algorithm are used in a real-time simulation of the water distribution system. The resilient ℓ_2 observer in (38) is included to perform the state estimation. The goal of this simulation is to estimate the probability of achieving resilient performance, using different attack detection strategies with the ℓ_2 observer. Thus, a Monte-Carlo experiment is carried out. The input of the experiment are generated attacks by the trained attack generator. The outputs are: (1) the **probability of sustaining normal operation** (pSNO) estimated as the ratio of the instances where the deviation of critical measurement from the nominal value is below a given threshold to the total number of instances, (2) the **probability of triggering an alarm** (pTA) estimated as the ratio of the instances where the deviation of detection residual from the nominal value is beyond a given threshold to the total number of instances. Clearly, bigger values of pSNO and pTA indicate better resiliency.

The input attack dataset includes different number of attacks from $1(2.5\%)$ to $20(50\%)$, and for each number of attacks, we generated 20 datasets corresponding to different attack scenarios. Successful attacks could not be found when the number of attacks is equal to 1. Thus, that case is excluded. For the remaining cases, with the number of attacks ranging from 2 to 20, attack supports were generated randomly.

Next, the system simulation process shown in Fig. 2 was carried out with different attack detection strategies: MLP1, MLP1 with pruning, MLP2, and MLP2 with pruning. During the system simulation, the T-horizon cumulative detection residual $\mathbf{r}_I$ is calculated based on (14), and the critical measurement $\mathbf{y}_{c_{i+1}}$ is also calculated based on (16). Then the performance, in terms of deviation ratios of the residual $\Delta\mathbf{r}$ and the critical measurements $\Delta\mathbf{y}$ respectively, are evaluated using

$$\Delta\mathbf{r} = \frac{|(\mathbf{r}_I - \mathbf{r}_d) - (\mathbf{r}_I^\star - \mathbf{r}_d)|}{\max\left(|(\mathbf{r}_I - \mathbf{r}_d) - (\mathbf{r}_I^\star - \mathbf{r}_d)|\right)},$$

$$\Delta\mathbf{y} = \frac{\left|(\mathbf{y}_{c_{i+1}} - \mathbf{y}_{c_d}) - (\mathbf{y}^\star_{c_{i+1}} - \mathbf{y}_{c_d})\right|}{\max\left(\left|(\mathbf{y}_{c_{i+1}} - \mathbf{y}_{c_d}) - (\mathbf{y}^\star_{c_{i+1}} - \mathbf{y}_{c_d})\right|\right)},$$

where $\mathbf{r}_d$ is the residual vector in the corresponding attack-free and noise-free case, $\mathbf{y}_{c_d} = C_c\mathbf{x}_d$ is the desired critical measurement vector corresponding to the desired states $\mathbf{x}_d$. Consequently, pTA and pSNO are estimated as $\text{pTA} = \frac{1}{N}\left\|\Delta\mathbf{r} \geq \frac{\tau_1|\mathbf{y}_c^\star|}{\max(|\mathbf{y}_c - \mathbf{y}_c^\star|)}\right\|_{\ell_0}$ and $\text{pSNO} = \frac{1}{N}\left\|\Delta\mathbf{y} \leq \frac{\tau_2|\mathbf{r}^\star|}{\max(|\mathbf{r} - \mathbf{r}^\star|)}\right\|_{\ell_0}$ respectively, where N is the total number of time instances from the start of attack injection.

The thresholds $\tau_1 = 1\%$ and $\tau_2 = 3\%$ are set based on the responses of the attack-free simulation.

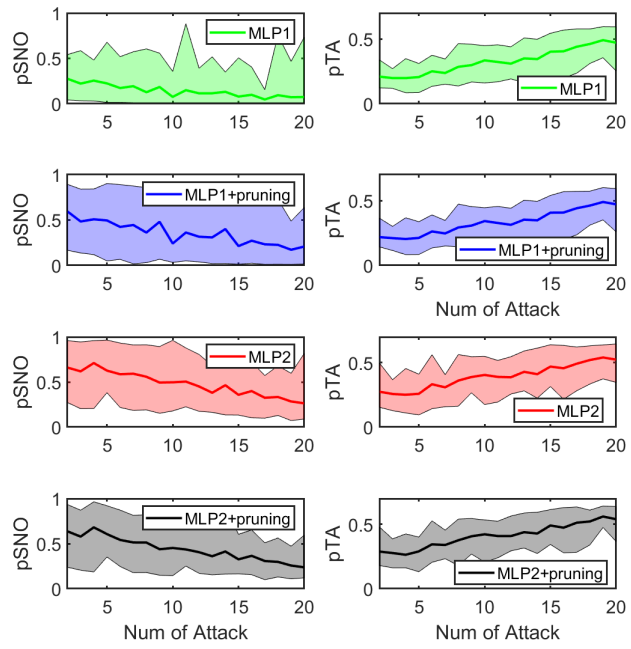

Fig. 8. A comparison of the ℓ_2 observer resiliency with different attack detection strategies, bigger values of pSNO and pTA indicate better resiliency. (The solid lines are mean values and the colored region indicates the spread of quantities

Fig. 8 shows the result of the experiment and a comparison of pSNO and pTA for different detection strategies. It is seen that MLP2 has better resiliency than MLP1, even better than MLP1 with pruning. This demonstrates the significant improvement in resiliency by including the automated attack generator in the training of MLP. In addition, the mean of localization precision, calculated by (26), of four attack detection strategies are shown in Fig. 9. As discussed in the last subsection, although the pruning algorithm still improve the localization precision of MLP2, the state estimation result is not improved as much due to the loss of observability. This indicates the MLP2 has already achieved the best balance between the localization precision and the observability requirement. However, it is noteworthy that MLP2 with pruning has tighter spread of pSNO and pTA values than MLP2. This indicates pruning algorithm can reduce performance uncertainty.

Finally, one time-domain simulation result is presented in Fig. 10. The attacks, generated by the trained generator in Fig. 1, are injected through the channels $\mathcal{T} = \{2, 8\}$[1], and four resilient estimation schemes are compared. It is seen that the schemes using MLP2 and MLP2 with pruning reduce the effectiveness of the critical measurements (smaller $\Delta\mathbf{y}$), and also reduce the stealthiness of detection (bigger $\Delta\mathbf{r}$). Notice that the peak value of $\Delta\mathbf{y}$ at the beginning of injecting attacks

[1]The measurement channel used were not optimized or chosen using any systematic way. The main objective of this simulation was not to demonstrate how big of a difference MLP2 can make but to see what the time-domain performance signals look like for one case of the stochastic experiment presented in Fig. 8

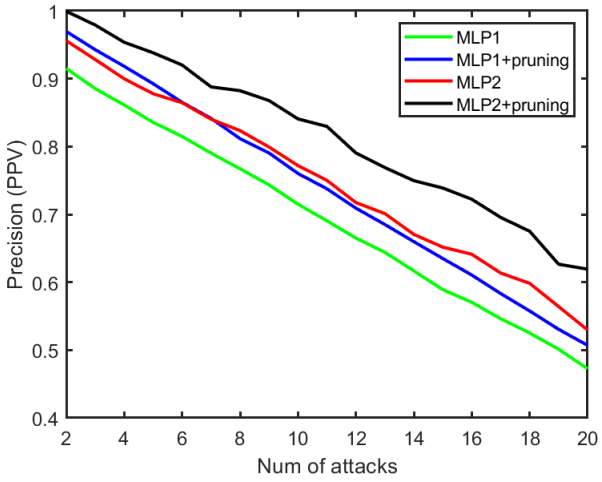

Fig. 9. A comparison of the mean precision of different attack detection strategies in the system simulation versus different number of attacks

happens for all four localization strategies, it is because of the lack of enough measurement history for MLP.

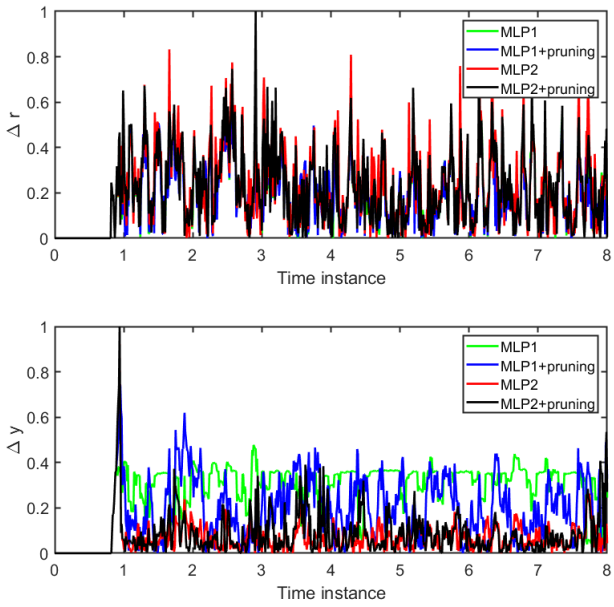

Fig. 10. An example simulation results of the four different attack localization strategies under 2 attacks at $\mathcal{T} = \{2, 8\}$.

VIII. Conclusion

In this paper, we present an algorithm design for resilient cyber-physical system, the resiliency is significantly improved by including the proposed automated attack generation in the training of attack detection algorithm. Unlike traditional GAN-based FDI attack generation, the automated attack generator does not require the prepared attack samples.

IX. Acknowledgements

This material is based upon work supported by the Department of Energy under Award Number DE-CR0000005

REFERENCES

[1] Liang *et al.*, "The 2015 ukraine blackout: Implications for false data injection attacks," *IEEE Transactions on Power Systems*, vol. 32, no. 4, pp. 3317–3318, 2016.

[2] R. M. Clark, S. Panguluri, T. D. Nelson, and R. P. Wyman, "Protecting drinking water utilities from cyberthreats," *Journal of the American Water Works Association*, vol. 109, no. INL/JOU-16-39302, 2017.

[3] B. Brentan, P. Rezende, D. Barros, G. Meirelles, E. Luvizotto, and J. Izquierdo, "Cyber-attack detection in water distribution systems based on blind sources separation technique," *Water*, vol. 13, no. 6, p. 795, 2021.

[4] Y. Liu, P. Ning, and M. K. Reiter, "False data injection attacks against state estimation in electric power grids," *ACM Transactions on Information and System Security (TISSEC)*, vol. 14, no. 1, p. 13, 2011.

[5] R. B. Bobba, K. M. Rogers, Q. Wang, H. Khurana, K. Nahrstedt, and T. J. Overbye, "Detecting false data injection attacks on dc state estimation," in *Preprints of the First Workshop on Secure Control Systems, CPSWEEK*, vol. 2010, 2010.

[6] K. C. Sou *et al.*, "Electric power network security analysis via minimum cut relaxation," in *Decision and Control and European Control Conference, 50th IEEE Conference on*. IEEE, 2011, pp. 4054–4059.

[7] M. Kordestani and M. Saif, "Observer-based attack detection and mitigation for cyberphysical systems: A review," *IEEE Systems, Man, and Cybernetics Magazine*, vol. 7, no. 2, pp. 35–60, 2021.

[8] A. Majumdar and B. C. Pal, "Bad data detection in the context of leverage point attacks in modern power networks," *IEEE Transactions on Smart Grid*, vol. 9, no. 3, pp. 2042–2054, 2016.

[9] B. Tang, J. Yan, S. Kay, and H. He, "Detection of false data injection attacks in smart grid under colored gaussian noise," in *2016 IEEE Conference on Communications and Network Security (CNS)*. IEEE, 2016, pp. 172–179.

[10] Li *et al.*, "Detecting false data injection attacks against power system state estimation with fast go-decomposition (godec) approach," *IEEE Transactions on Industrial Informatics*, 2018.

[11] B. Li, R. Lu, and G. Xiao, "Hmm-based fast detection of false data injections in advanced metering infrastructure," in *GLOBECOM 2017-2017 IEEE Global Communications Conference*. IEEE, 2017, pp. 1–6.

[12] D. Ding, Q.-L. Han, Y. Xiang, X. Ge, and X.-M. Zhang, "A survey on security control and attack detection for industrial cyber-physical systems," *Neurocomputing*, vol. 275, pp. 1674–1683, 2018.

[13] D. B. Rawat and C. Bajracharya, "Detection of false data injection attacks in smart grid communication systems," *IEEE Signal Processing Letters*, vol. 22, no. 10, pp. 1652–1656, 2015.

[14] Chaojun *et al.*, "Detecting false data injection attacks in ac state estimation," *IEEE Transactions on Smart Grid*, vol. 6, no. 5, pp. 2476–2483, 2015.

[15] N. ŽIVKOVIĆ and A. T. SARIĆ, "Detection of false data injection attacks using unscented kalman filter," *Journal of Modern Power Systems and Clean Energy*, pp. 1–13, 2018.

[16] Y. Mo and B. Sinopoli, "False data injection attacks in control systems," in *Preprints of the 1st workshop on Secure Control Systems*, 2010, pp. 1–6.

[17] Y. Zheng and O. M. Anubi, "Attack-resilient weighted ℓ_1 observer with prior pruning," in *2021 American control conference*. IEEE, 2021.

[18] M. Ozay *et al.*, "Machine learning methods for attack detection in the smart grid," *IEEE transactions on neural networks and learning systems*, vol. 27, no. 8, pp. 1773–1786, 2016.

[19] O. Boyaci, M. R. Narimani, K. Davis, M. Ismail, T. J. Overbye, and E. Serpedin, "Joint detection and localization of stealth false data injection attacks in smart grids using graph neural networks," *arXiv preprint arXiv:2104.11846*, 2021.

[20] Esmalifalak *et al.*, "Stealth false data injection using independent component analysis in smart grid," in *SmartGridComm, IEEE International Conference on*. IEEE, 2011, pp. 244–248.

[21] J. Yan *et al.*, "Detection of false data attacks in smart grid with supervised learning," in *Neural Networks (IJCNN), 2016 International Joint Conference on*. IEEE, 2016, pp. 1395–1402.

[22] D. Wilson, Y. Tang, J. Yan, and Z. Lu, "Deep learning-aided cyber-attack detection in power transmission systems," in *2018 IEEE Power & Energy Society General Meeting (PESGM)*. IEEE, 2018, pp. 1–5.

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

[24] S. C. Lee and D. V. Heinbuch, "Training a neural-network based intrusion detector to recognize novel attacks," *IEEE Transactions on systems, man, and Cybernetics-Part A: Systems and Humans*, vol. 31, no. 4, pp. 294–299, 2001.

[25] D. Ding, Q.-L. Han, X. Ge, and J. Wang, "Secure state estimation and control of cyber-physical systems: A survey," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 51, no. 1, pp. 176–190, 2020.

[26] H. Fawzi, P. Tabuada, and S. Diggavi, "Secure estimation and control for cyber-physical systems under adversarial attacks," *IEEE Transactions on Automatic control*, vol. 59, no. 6, pp. 1454–1467, 2014.

[27] M. Pajic, J. Weimer, N. Bezzo, P. Tabuada, O. Sokolsky, I. Lee, and G. J. Pappas, "Robustness of attack-resilient state estimators," in *2014 ACM/IEEE International Conference on Cyber-Physical Systems (ICCPS)*. IEEE, 2014, pp. 163–174.

[28] M. Pajic, P. Tabuada, I. Lee, and G. J. Pappas, "Attack-resilient state estimation in the presence of noise," in *2015 54th IEEE Conference on Decision and Control (CDC)*. IEEE, 2015, pp. 5827–5832.

[29] C. Lee, H. Shim, and Y. Eun, "Secure and robust state estimation under sensor attacks, measurement noises, and process disturbances: Observer-based combinatorial approach," in *2015 European Control Conference (ECC)*. IEEE, 2015, pp. 1872–1877.

[30] Y. Shoukry and P. Tabuada, "Event-triggered state observers for sparse sensor noise/attacks," *IEEE Transactions on Automatic Control*, vol. 61, no. 8, pp. 2079–2091, 2015.

[31] M. S. Chong, M. Wakaiki, and J. P. Hespanha, "Observability of linear systems under adversarial attacks," in *2015 American Control Conference (ACC)*. IEEE, 2015, pp. 2439–2444.

[32] D. Ding, Z. Wang, Q.-L. Han, and G. Wei, "Security control for discrete-time stochastic nonlinear systems subject to deception attacks," *IEEE Transactions on Systems, Man, and Cybernetics: Systems*, vol. 48, no. 5, pp. 779–789, 2016.

[33] W. Wan, H. Kim, N. Hovakimyan, and P. G. Voulgaris, "Attack-resilient estimation for linear discrete-time stochastic systems with input and state constraints," in *2019 IEEE 58th Conference on Decision and Control (CDC)*. IEEE, 2019, pp. 5107–5112.

[34] O. M. Anubi and C. Konstantinou, "Enhanced resilient state estimation using data-driven auxiliary models," *IEEE Transactions on Industrial Informatics*, vol. 16, no. 1, pp. 639–647, 2019.

[35] O. M. Anubi, C. Konstantinou, and R. Roberts, "Resilient optimal estimation using measurement prior," *arXiv preprint arXiv:1907.13102*, 2019.

[36] O. M. Anubi, C. Konstantinou, C. A. Wong, and S. Vedula, "Multi-model resilient observer under false data injection attacks," in *2020 IEEE Conference on Control Technology and Applications (CCTA)*. IEEE, 2020, pp. 1–8.

[37] Y. Zheng and O. M. Anubi, "Attack-resilient observer pruning for path-tracking control of wheeled mobile robot," in *2020 ASME Dynamic Systems and Control (DSC) Conference*. ASME, 2020, pp. 1–9.

[38] Q. Yang, J. Yang, W. Yu, D. An, N. Zhang, and W. Zhao, "On false data-injection attacks against power system state estimation: Modeling and countermeasures," *IEEE Transactions on Parallel and Distributed Systems*, vol. 25, no. 3, pp. 717–729, 2013.

[39] M. Mohammadpourfard, F. Ghanaatpishe, M. Mohammadi, S. Lakshminarayana, and M. Pechenizkiy, "Generation of false data injection attacks using conditional generative adversarial networks," in *2020 IEEE PES Innovative Smart Grid Technologies Europe (ISGT-Europe)*. IEEE, 2020, pp. 41–45.

[40] S. Ahmadian, H. Malki, and Z. Han, "Cyber attacks on smart energy grids using generative adverserial networks," in *2018 IEEE Global Conference on Signal and Information Processing (GlobalSIP)*. IEEE, 2018, pp. 942–946.

[41] M. Fernández and S. Williams, "Closed-form expression for the poisson-binomial probability density function," *IEEE Transactions on Aerospace and Electronic Systems*, vol. 46, no. 2, pp. 803–817, 2010.

[42] A. Alessandri, M. Baglietto, and G. Battistelli, "Receding-horizon estimation for discrete-time linear systems," *IEEE Transactions on Automatic Control*, vol. 48, no. 3, pp. 473–478, 2003.

[43] I. Goodfellow, J. Pouget-Abadie, M. Mirza, B. Xu, D. Warde-Farley, S. Ozair, A. Courville, and Y. Bengio, "Generative adversarial nets," *Advances in neural information processing systems*, vol. 27, 2014.

[44] G. Cybenko, "Approximation by superpositions of a sigmoidal function," *Mathematics of control, signals and systems*, vol. 2, no. 4, pp. 303–314, 1989.

[45] T. Fawcett, "An introduction to roc analysis," *Pattern recognition letters*, vol. 27, no. 8, pp. 861–874, 2006.