Design of Switching Attacks with Physics-Informed Neural Networks

Alex Parri, Qiuhan Cao, and Francesco Liberati*.

***Abstract*— This paper proposes a nonlinear optimal switching attack for Cyber-Physical Systems based on Physics-Informed Neural Networks (PINNs). The methodology employs a two-stage training strategy to identify optimal attack sequences: an attack discovery phase that temporarily relaxes physical constraints to explore divergent system behaviors, and a physics refining phase that enforces dynamical consistency to ensure feasibility. Beyond the standard physics loss, the objective function incorporates an attractivity term to bias the search toward divergent trajectories and a damage term to maximize angular velocity. This formulation enables the network to prioritize attack impact while maintaining physical plausibility. Unlike existing methods limited by linearized models or predefined sliding surfaces, the proposed approach automatically learns high-impact strategies directly from the inherent nonlinear system dynamics.**

I. INTRODUCTION

Cyber-Physical Systems (CPSs), as the deep integration of cyberspace and the physical space, have been widely deployed in critical domains such as energy, transportation, and industrial control. While the deep integration of information systems has enhanced CPSs control and intelligence, it also increases their exposure to cyber-attacks through open and complex communication technologies, with attacks capable of causing physical damage [1]. A well-known case is the "Aurora" attack experiment conducted in the United States in 2007. This attack manipulated circuit breakers at specific intervals, ultimately resulting in permanent physical damage to the equipment. This experiment was one of the first to demonstrate that cyberattacks could inflict tangible physical destruction in the real world, revealing inherent security flaws of CPSs, whose architecture often prioritize accessibility and operational efficiency, relegating cybersecurity to a secondary consideration [2]. Therefore, this paper focuses on switching attacks, a class of cyber-physical attacks designed to cause severe physical damage through the malicious manipulation of discrete control actions in physical components, leading to direct disruption of system dynamics.

The conceptual analysis and design framework for switching attacks were first introduced for the security of the power system in [3], [4], [5], [6] by Kundur et al. Specifically, reference [3] introduced a class of state-dependent coordinated switching attacks, designed using the theory of variable structure systems. A framework for ranking switch vulnerability based on a linearized model was proposed in [4], [5], with [5] showing that the devised attacks retain efficacy against nonlinear systems despite model errors and partial state information. Further research in [6] addressed the performance degradation experienced by sliding-mode switching attacks in non-ideal cyber environments. The reliance on linearized models means that for strongly nonlinear CPSs with wide-range dynamics, the designed attack strategies may be suboptimal or even ineffective.

To overcome the limitations of linear methods, Aradhna et al. [7] proposed a non-linear sliding surface that utilized the swing equation to identify stability boundaries to generate switching sequences. The study demonstrated that this method reduced chattering and shortened the attack duration compared to traditional linear sliding surfaces. Falah et al. [8] proposed a multi-stage load oscillation attack, which was initiated by learning the line flow sensitivity factors through small load disturbances and adjusting the oscillation frequency in real time. Switching attacks also have the potential to launch large-scale, covert physical damage in emerging infrastructure. For example, [9] utilized public data for system identification to generate attack commands, and [10] designed switching attacks based on the eigenvalue algorithm to control the charging and discharging of electric vehicles. The effectiveness of both attack strategies hinges on acquiring an accurate model of the system.

Given the limitations of existing research, this work proposes a nonlinear optimal switching attack method using Physics-Informed Neural Networks (PINN). Unlike conventional approaches, this method automatically learns optimal attack through offline PINN training, enabling the discovery of switching strategies that would be difficult or computationally prohibitive to derive by hand in nonlinear systems.

The remainder of the paper is structured as follows. The system and PINN-based model description are presented in Section II. Section III introduces the proposed PINN-based attack design method. Section IV presents the simulation results. Finally, Section V concludes this work.

II. MODEL DESCRIPTION

A. System Modeling

This work simplifies the overall system through the Single Machine Infinite Bus (SMIB) model, shown in Fig. 1, consisting of a single target generator G_i to which the load P_L is connected through a single target switch S_L, while the rest of the grid is grouped within the term G_∞. This SMIB

Alex Parri is with the Department of Information Engineering, Computer Science and Statistics, Sapienza University of Rome, 00185 Rome, Italy (e-mail: parri.2140961@studenti.uniroma1.it). Qiuhan Cao, and Francesco Liberati are with the Department of Computer, Control and Management Engineering, "Antonio Ruberti", Sapienza University of Rome, 00185 Rome, Italy (e-mail: {cao, liberati}@diag.uniroma1.it), (*Corresponding author: Francesco Liberati). This work is supported by the SAPIENZA - ATENEO 2025 "AI Methods for Cyber-physical Systems Security" project, no. RP125199C3A90E6B.

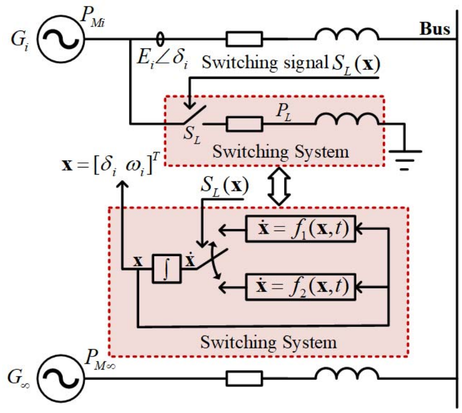


Fig. 1. Reference scenario with Single Machine Infinite Bus (SMIB) model

model can be expressed as

$$\begin{aligned} \dot{\delta}_i &= \omega_i \\ m_i\dot{\omega}_i &= P_{Mi} - d_i\omega_i - E_i^2 G_{ii} - P_{\max}\sin(\delta_i) - S_L(\mathbf{x})P_L \end{aligned} \tag{1}$$

where δ_i and ω_i denote the rotor angle and angular velocity deviation of G_i, respectively; m_i is the moment of inertia; P_{Mi} is the mechanical power of G_i; d_i is the damping coefficient; E_i and G_{ii} represent the internal voltage and conductance, respectively; $P_{\max}$ is a constant; and $\mathbf{x} = [\delta_i \quad \dot{\delta}_i]^T$.

Switch S_L is a function of sliding surface $s(\mathbf{x})$ representing a binary switch, ideally represented by a Heaviside step function $H(\mathbf{x})$, which would ensure that the load is zero when the switch is closed and P_L when it is open

$$S_L(\mathbf{x}) = H(s(\mathbf{x})) = \begin{cases} 1 & s(\mathbf{x}) \geq 0 \\ 0 & s(\mathbf{x}) < 0. \end{cases} \tag{2}$$

Rewriting (1) explicitly in terms of δ_i yields:

$$m_i\ddot{\delta}_i + d_i\dot{\delta}_i + E_i^2 G_{ii} + P_{\max}\sin(\delta_i) - P_{Mi} + S_L(\mathbf{x})P_L = 0, \tag{3}$$

define the acceleration $\ddot{\delta}_i' = -\frac{d_i\dot{\delta}_i - E_i^2 G_{ii} - P_{max}\sin(\delta_i) + P_{Mi}}{m_i}$ by isolating $\ddot{\delta}_i$ from (3) without the load term P_L. So the system (1) takes the form

$$\begin{aligned} \dot{\delta}_i &= \frac{d}{dt}\delta_i \\ \ddot{\delta}_i &= \left\{ \begin{array}{ll} \ddot{\delta}_i{}' - \frac{P_L}{m_i} & s(\mathbf{x}) \geq 0 \\ \ddot{\delta}_i{}' & s(\mathbf{x}) < 0 \end{array} \right\}. \end{aligned} \tag{4}$$

Knowing that $\dot{\delta}_i$ is always defined as $\frac{d}{dt}\delta_i$, the variable part is only present in $\ddot{\delta}_i$. Therefore, it can be formulated in the standard form of variable structure system theory:

$$\ddot{\delta}_i = \begin{cases} f_1(\mathbf{x},t) = \ddot{\delta}_i{}' - \frac{P_L}{m_i} & s(\mathbf{x}) \geq 0 \\ f_2(\mathbf{x},t) = \ddot{\delta}_i{}' & s(\mathbf{x}) < 0. \end{cases} \tag{5}$$

As shown in Fig. 1, through the behavior of switch S_L, the switching system can alternate between two different structures, $f_1(\mathbf{x},t)$ and $f_2(\mathbf{x},t)$, thereby forming the variable structure system as described in (5).

Lemma 1 (Reaching condition): Given the variable structure system

$$\dot{\mathbf{x}} = \begin{cases} f_1(\mathbf{x},t) & s(\mathbf{x}) > 0 \\ f_2(\mathbf{x},t) & s(\mathbf{x}) < 0 \end{cases} \tag{6}$$

with $\dot{s}(\mathbf{x}) \neq 0$, a sufficient condition for the trajectory to reach the switching surface $s(\mathbf{x}) = 0$ is

$$s(\mathbf{x})\dot{s}(\mathbf{x}) < 0 \implies \begin{cases} \dot{s}(\mathbf{x}) < 0 & \text{if } s(\mathbf{x}) > 0 \\ \dot{s}(\mathbf{x}) > 0 & \text{if } s(\mathbf{x}) < 0. \end{cases} \tag{7}$$

This implies that an attacker controlling $s(\mathbf{x})$ can steer the trajectory toward the switching surface, thereby allowing for exploitation of system modes $f_i(\mathbf{x},t)$ such that the trajectory moves simultaneously towards the surface and away from the origin, achieving divergence of state variables $\mathbf{x}$.

B. PINN Modeling

Artificial neural networks derive their power from a fundamental unit called the perceptron. A typical feedforward network is composed of l layers, each built using an array of an arbitrary number n_i of perceptrons. Let $f^{(i)}(\cdot)$ denote the activation function of layer i. The output $o_i(x)$ of layer i is then given by

$$o_i(x) = f^{(i)}(\mathbf{W}_i^T \mathbf{x}_{i-1} + \mathbf{b}_i) \tag{8}$$

where weights $\mathbf{W}_i = \begin{bmatrix} w_{1,1}^{(i)} & w_{1,2}^{(i)} & \cdots & w_{1,n_{i-1}}^{(i)} \\ w_{2,1}^{(i)} & w_{2,2}^{(i)} & \cdots & w_{2,n_{i-1}}^{(i)} \\ \vdots & \vdots & \ddots & \vdots \\ w_{n_i,1}^{(i)} & w_{n_i,2}^{(i)} & \cdots & w_{n_i,n_{i-1}}^{(i)} \end{bmatrix}^T$, the previous layer's outputs $\mathbf{x}_{i-1} = \begin{bmatrix} x_1^{(i-1)} & x_2^{(i-1)} & \cdots & x_{n_{i-1}}^{(i-1)} \end{bmatrix}^T$, and bias terms $\mathbf{b}_i = \begin{bmatrix} b_1^{(i)} & b_2^{(i)} & \cdots & b_{n_i}^{(i)} \end{bmatrix}^T$.

Traditional artificial neural networks can approximate arbitrary functions, yet they often struggle to model physical systems due to their disregard for the underlying differential equations, leading to overfitting and poor generalization [11]. To address this, this work employs PINN, which are designed to learn functions governed by

$$f(x,t) = \partial u(x,t) + \mathcal{N}[u(x,t),\lambda] = 0 \quad x \in \Omega, t \in [0,T] \tag{9}$$

where $u(x,t)$ is the solution of equation, $\mathcal{N}[\cdot]$ is a non-linear differential operator expressing the relation between u and possibly unknown system parameters λ.

The solution is learned by minimizing a loss function

$$\mathcal{L}(\mathbf{x}) = \lambda_{data}\mathcal{L}_{data}(\mathbf{x}) + \lambda_{phy}\mathcal{L}_{phy}(\mathbf{x}) + \lambda_{ic}\mathcal{L}_{ic}(\mathbf{x}) \tag{10}$$

where the three loss terms are all defined as Mean Squared Errors (MSEs). Specifically, the data loss $\mathcal{L}_{data}(\mathbf{x}) = \frac{1}{N_u}\sum_{i=1}^{N_u}||u(x_u^i,t_u^i) - u^i||^2$ evaluates the fit to N_u given boundary data points; the physics loss $\mathcal{L}_{phy}(\mathbf{x}) = \frac{1}{N_f}\sum_{i=1}^{N_f}||f(x_f^i,t_f^i)||^2$ enforces the residuals of the governing equation (9) at N_f collocation points; and the initial condition loss $\mathcal{L}_{ic}(\mathbf{x}) = \frac{1}{N_0}\sum_{i=1}^{N_0}||u(x_0^i,0) - u_0^i||^2$ corresponds to the N_0 initial condition points.

III. PROPOSED PINN-BASED ATTACK

A. Proposed PINN-Based Attack Design

The attack is obtained by training and subsequently querying a PINN $f(\mathbf{x}_0, \mathbf{t})$, where $\mathbf{x}_0 = [\delta_0 \quad \dot{\delta}_0]^T$ is the initial condition vector and $\mathbf{t} = [t_0 \quad t_1 \quad \cdots \quad t_f]^T$ is a time vector containing all time instances from $t_0 = 0$ to $t_f = T$, the final time. The network then outputs an array $\boldsymbol{\delta}$ containing every δ_i for each time t_i and the corresponding value of the switching function $\mathbf{s}(\mathbf{x}) = \mathbf{s}(\{\boldsymbol{\delta}, \dot{\boldsymbol{\delta}}\})$

$$f(\mathbf{x}_0, \mathbf{t}) \rightarrow \{\boldsymbol{\delta}, \mathbf{s}\}. \tag{11}$$

Before any attack could be found, the network would first need to correctly simulate the system dynamics. This is obtained by imposing the PINN $f(\mathbf{x}_0, \mathbf{t})$ equal to the previously shown single variable SMIB (3):

$$\begin{aligned} f(\mathbf{x}_0, \mathbf{t}) =& m_i \boldsymbol{\delta}_{tt} + d_i \boldsymbol{\delta}_t + P_{\max} \sin(\boldsymbol{\delta}) \\ &+ (E_i^2 G_{ii} - P_{Mi} + S_L(\mathbf{s}) P_{Li}) \cdot \mathbf{1} \end{aligned} \tag{12}$$

and

$$S_L(\mathbf{s}) = \frac{\arctan(100 \cdot \mathbf{s})}{\pi} + \frac{1}{2} \tag{13}$$

which $\boldsymbol{\delta}_t = [\dot{\delta}_0 \quad \dot{\delta}_1 \quad \cdots \quad \dot{\delta}_f]^T$, $\boldsymbol{\delta}_{tt} = [\ddot{\delta}_0 \quad \ddot{\delta}_1 \quad \cdots \quad \ddot{\delta}_f]^T$, and $\mathbf{1} = [1, 1, \ldots, 1]^T \in \mathbb{R}^{N_t}$ a vector of ones with the same dimension as $\mathbf{t}$ (N_t is the number of time points).

Remark 1: The Heaviside step function (2) is discontinuous and non-differentiable at zero. It is difficult to use it directly in numerical optimization and continuous control. Therefore, this paper adopts a smooth and differentiable saturation function (13), which achieves a smooth transition from 0 to 1 via the arctangent function, offering good continuity and differentiability. The heuristic scaling factor 100 controls the steepness of the transition, allowing the function to approximate the ideal step-like behavior while maintaining numerical stability and practical feasibility.

In addition to the above, the network is provided with the same initial condition vector $\mathbf{x}_0$ within the loss function:

$$\begin{aligned} &\mathscr{L}(\mathbf{x}) = \\ &\lambda_{phy}\mathscr{L}_{phy}(\mathbf{x}) + \lambda_{attr}\mathscr{L}_{attr}(\mathbf{x}) + \lambda_{dmg}\mathscr{L}_{dmg}(\mathbf{x}) + \lambda_{ic}\mathscr{L}_{ic}(\mathbf{x}) \end{aligned} \tag{14}$$

where

$$\begin{aligned} \mathscr{L}_{phy}(\mathbf{x}) &= \frac{1}{N_f} \sum_{i=1}^{N_f} ||f(x_f^i, t_f^i)||^2 \\ \mathscr{L}_{attr}(\mathbf{x}) &= \frac{1}{N_f} \sum_{i=1}^{N_f} \mathrm{ReLU}(s^i \cdot s_t^i), \quad \mathrm{ReLU}(x) \triangleq \max(0, x) \\ \mathscr{L}_{dmg}(\mathbf{x}) &= -\frac{1}{N_f} \sum_{i=1}^{N_f} ||u_t^i||^2 \\ \mathscr{L}_{ic}(\mathbf{x}) &= \frac{1}{N_0} \sum_{i=1}^{N_0} ||u(x_0^i, 0) - u_0^i||^2 \end{aligned} \tag{15}$$

in which $\mathscr{L}_{phy}$ and $\mathscr{L}_{ic}$ are the standard PINN losses as seen in (10); the attractivity loss $\mathscr{L}_{attr}$ attempts to have the model respect during the attack discovery phase the sliding mode condition previously mentioned in (7), where the ReLU ensures that the component is kept below zero as long as possible; and $\mathscr{L}_{dmg}$ aids the model onto pushing away from the origin, and is defined as the negative square norm of ω.

Remark 2: Unlike the general PINN loss function (10), the above loss does not include any supervised data component: it was decided this way to avoid the network being biased to a specific solution of the SMIB equation (1) and thus not being able to generalize in the space of possible switching attacks.

B. Data Training

To search for the optimal attack, the neural network adopts the model architecture from TABLE I. The network consists of four layers, each containing 55 neurons. All layers employ the sine function as the activation function, with the exception of output layer being a linear layer. Provided how strongly non-linear is the equation the network is trying to learn (11), the choice of sine function as activation function is effective as it yields to better generalization and lower overfitting [12]. The overall network has 9572 parameters, including all zero bias parameters.

TABLE I
MODEL STRUCTURE FOR THE OPTIMAL ATTACK DISCOVERY

Layer	**Activation** $f^{(i)}(\cdot)$	**Output shape**	**Parameters**
Sine	$\sin(3 \cdot \mathbf{W}^T\mathbf{x} + \mathbf{b})$	$[N_f, 55]$	220
Sine	$\sin(3 \cdot \mathbf{W}^T\mathbf{x} + \mathbf{b})$	$[N_f, 55]$	3080
Sine	$\sin(3 \cdot \mathbf{W}^T\mathbf{x} + \mathbf{b})$	$[N_f, 55]$	3080
Linear	$\mathbf{W}^T\mathbf{x} + \mathbf{b}$	$[N_f, 2]$	112

Prior to training, $N_{\text{states}} = 10\,000$ unique initial state vectors $\mathbf{x}_{0,i}$ are uniformly sampled within $[-1.0, 1.0]^2$, with a minimum spacing $\Delta x_0 = 0.01$, and stored in a matrix $\mathbf{X}_0$. $N_f = 8000$ physics collocation points are used, following the optimal trade-off between accuracy and training time suggested in [13]. The weights W_i of the sine-activated network are initialized by sampling from a uniform distribution $w_{ij} \sim U(-\sigma, \sigma)$ with standard deviation σ as follows:

$$\sigma = \begin{cases} \frac{1}{n_{in}} & , \quad \text{Input layer} \\ \sqrt{\frac{6}{n_{in} w_0}} \frac{1}{\sqrt{n_{in} w_0}} & , \quad \text{Hidden layer} \end{cases} \tag{16}$$

where n_{in} is the number of incoming connections to layer i, $b_i = 0$. Weights w_0 and b directly represent angular frequency and phase offset respectively. This initialization is proven to speed up convergence by a small margin through better learning of high frequency oscillation components [12].

At the beginning of each epoch j, a batch Θ_j of size $N_{\text{batch}} = 1024$ is constructed by uniformly sampling from $\mathbf{X}_0$ and the time vector $\mathbf{t}$:

$$\Theta_j = \{(\mathbf{x}_{0,i}^{(j)}, \mathbf{t}_i^{(j)})_{i=1}^{1024}\} \leftarrow \begin{cases} \mathbf{x}_0^{(j)} \leftarrow \$_{1024}\mathbf{X}_0 \\ \mathbf{t}^{(j)} \leftarrow \$_{1024}\mathbf{t} \end{cases} \tag{17}$$

where the $\leftarrow \$_i$ notation randomly selects i distinct elements from the set, without repetition. This batch Θ_j is then input to the PINN for training. Consequently, the network processes only $N_f^{\text{train}} = 1024$ physics collocation points per epoch, which is significantly fewer than the total $N_f = 8000$ points.

TABLE II

ADAM HYPERPARAMETER CONFIGURATION

Learning rate	First moment	Second moment	Weight decay
$\gamma_{\text{init}} = 0.0075$	$\beta_1 = 0.95$	$\beta_2 = 0.99$	$\lambda = 10^{-5}$

The training employs the Adam optimizer with hyperparameters specified in Table II. The learning rate is reduced by a factor of 0.7 upon plateau detection, using a patience of 500 epochs, until reaching a minimum of $\gamma_{\min} = 5 \times 10^{-6}$.

Following a curriculum learning strategy, training proceeds in two stages over N_{epoch} epochs:

- Attack discovery (first N_{attack} epochs): Loss weights configured according to (14).
- Physics refining (remaining epochs): The weight on the physics term λ_{phy} is increased to its maximum value, while the damage weight λ_{dmg} is reduced to zero:

$$\lambda_{phy} \to 1 \quad \lambda_{dmg} \to 0. \tag{18}$$

The idea of the network is that of initially trading the physics accuracy to try and discover quicker diverging attacks, afterwards when this attack is discovered the physics leading to it are improved to output a more precise result. The core functioning lies in the fact that when the weight on damage λ_{dmg} is removed the network minimizes MSEs by sticking to the closest physically plausible attack rather than collapsing back to easier ones.

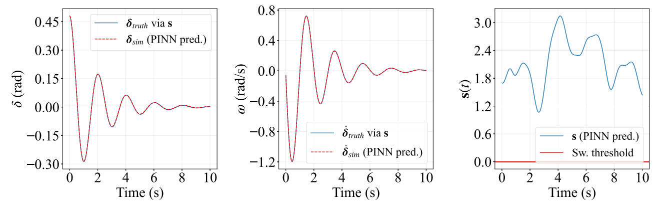
IV. SIMULATION RESULTS

Simulations have been tested on a computer with Ubuntu Desktop 24.04.3 LTS x86_64 running Python 3.12.3 code using the well-equipped PyTorch 2.8.0 library running on CUDA. A nonlinear model commonly adopted in the literature (e.g., [3], [5], [6]) is employed, with the simulation duration set to $T = 10\,\text{s}$:

$$\begin{aligned} \dot{\delta} &= \omega \\ \dot{\omega} &= -\omega - 10\sin(\delta) + 9 - 9S_L(\mathbf{s}). \end{aligned} \tag{19}$$

To make the simulation results more intuitive, the true state, $\dot{\mathbf{x}}_{truth} = \{\delta^{(i)}_{truth}, \omega^{(i)}_{truth}\}$, is obtained by feeding the output $\mathbf{s}$ from the neural network (12) into the switching function (13) and then calculating it using the system model (4); the simulated state, $\dot{\mathbf{x}}_{sim} = \{\delta, \dot{\delta}\} = \{\delta^{(i)}_{sim}, \omega^{(i)}_{sim}\}$, is taken directly from the neural network's output. The MSEs and average MSEs for the rotor angle δ and angular velocity ω are computed separately:

$$\begin{aligned} e_\delta &= \frac{1}{N_f}\sum_{i=1}^{N_f}\left(\delta^{(i)}_{truth} - \delta^{(i)}_{sim}\right)^2, \quad e^{avg}_\delta = \frac{1}{N_{test}}\sum_{i=1}^{N_{test}}\left(e^{(i)}_\delta\right)^2 \\ e_\omega &= \frac{1}{N_f}\sum_{i=1}^{N_f}\left(\omega^{(i)}_{truth} - \omega^{(i)}_{sim}\right)^2, \quad e^{avg}_\omega = \frac{1}{N_{test}}\sum_{i=1}^{N_{test}}\left(e^{(i)}_\omega\right)^2. \end{aligned} \tag{20}$$



(a) Rotor Angle (δ) (b) Angular Velocity (ω) (c) Switching Function

Fig. 2. Comparison of PINN prediction outputs and system model outputs

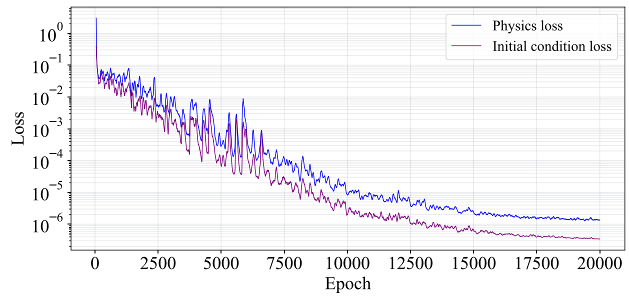


Fig. 3. Log-moving average of the physics loss and the initial condition loss over training epochs during system simulation

All simulations use $N_f = 8000$ physics collocation points, which corresponds to a sampling frequency of $f = 800\,\text{Hz}$. The forward propagation time of the network for a single output after training is approximately $(3.5 \pm 0.3)\,\text{ms}$.

A. System Simulation

This subsection evaluates the system performance without switching. Fig. 2 presents the comparisons between the PINN prediction outputs and the system model outputs for $N_{\text{test}} = 50$ initial states uniformly sampled from $[-1.5, 1.5]^2$, which extends the training domain $[-1.0, 1.0]^2$ by ± 0.5 in each dimension. No sample was selected more than once, including across training and testing phases. The average MSEs are:

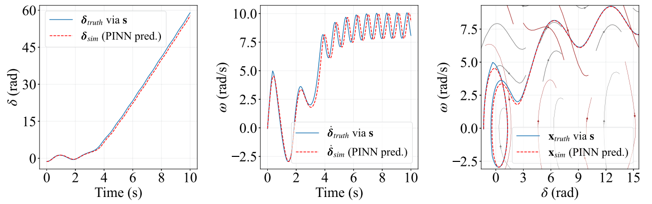
$$e^{\text{avg}}_\delta = 3.885 \times 10^{-4}, \quad e^{\text{avg}}_\omega = 2.252 \times 10^{-3}.$$

These results indicate that the PINN generalizes well to initial states outside the training distribution. Training was performed for $N_{\text{epochs}} = 20{,}000$ (2 min 12 s) with weights: $\lambda_{\text{phy}} = 1.0$, $\lambda_{\text{attr}} = \lambda_{\text{dmg}} = 0.0$, $\lambda_{\text{ic}} = 0.8$. The variation of the physics loss λ_{phy} and the initial condition loss λ_{ic} with the number of epochs is shown in Fig. 3.

B. Attack Execution

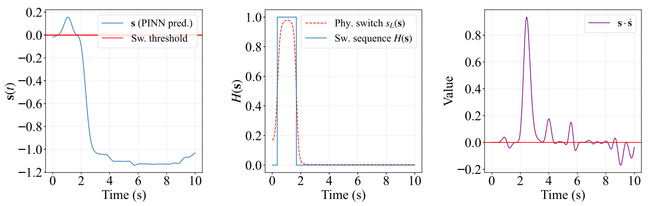
This subsection aims to leverage the trained neural network to discover optimal switching attacks. Data training results indicate that setting $N_{attack} = 20000$ yields the best balance between accuracy and attack optimality in all test scenarios. As shown in Fig. 4, under the initial condition $\mathbf{x}_0 = (-1.33, -0.07)$, the optimal attack identified by the network drives the system to instability. The corresponding MSEs are:

$$e_\delta = 0.818, \quad e_\omega = 0.573,$$



(a) Rotor Angle (δ) (b) Angular Velocity (ω) (c) Phase Portrait

Fig. 4. Physics system response under the optimal PINN-based switching attack, $\mathbf{x}_0 = (-1.33, -0.07)$



(a) Switching Function (b) Switching sequence (c) Attractivity condition

Fig. 5. Switching state of the optimal PINN-based switching attack, $\mathbf{x}_0 = (-1.33, -0.07)$

within a sample of $N_{\text{test}} = 50$ initial states, attacks are found in all cases, with average errors:

$$e_{\delta}^{\text{avg}} = 2.520, \quad e_{\omega}^{\text{avg}} = 1.32.$$

The details of the switching attack are presented in Fig. 5. Specifically, Fig. 5(a) shows that the network does not exploit low $\mathbf{s}$ values to solve $f(\mathbf{x}_0, \mathbf{t}) = 0$. Instead, once no further switches are needed to accelerate divergence, $\mathbf{s}$ drops negatively, relying mainly on $\boldsymbol{\delta}$. Fig. 5(b) shows that, despite initial discrepancies, the switching function $S_L(\mathbf{x})$ closely fits $H(\mathbf{x})$. Fig. 5(c) indicates that the attractivity condition $\mathbf{s}\cdot\dot{\mathbf{s}} < 0$ holds only transiently, indicating that the higher-impact discovered attack does not rely on a sliding mode. Figure 6 depicts the $\mathbf{s} = 0$ curve and its relation to the system trajectory in the phase plane, where the $\mathbf{s} = 0$ curve is a highly nonlinear, irregular green line. Near $\mathbf{x}_0$, the curve initially follows the trajectory before diverging, intersecting it exactly at the two switching instants.

Attack discovery used tuned weights: $\lambda_{\text{phy}} = 0.3$, $\lambda_{\text{attr}} = 0.05$, $\lambda_{\text{dmg}} = 0.1$, $\lambda_{\text{ic}} = 0.8$. Training lasted $N_{\text{epochs}} = 50000$ (5 min 31 s), with $N_{\text{attack}} = 20000$ epochs for attack discovery. As shown in Fig. 7, the physics loss and negative damage loss components increase steadily but do not diverge, whereas the initial condition loss oscillates around 10^{-2}. After attack discovery, the physics loss converges to 10^{-3} and the initial condition loss converges to 10^{-4}, while the negative damage converges to 5×10^2.

C. Performance of the Nonlinear Attack from Literature [7]

This subsection evaluates the proposed attack with that of [7], where the SMIB model was used as the first simulation scenario to ensure a fair and consistent evaluation under the same configuration. The function in [7] is defined as

$$\mathbf{s}_{paper}(\mathbf{x}) = K\mathbf{x} + \beta e^{\tanh(K\mathbf{x})} = 6\delta_1 + \omega_1 + 4e^{\tanh(6\delta_1 + \omega_1)}$$

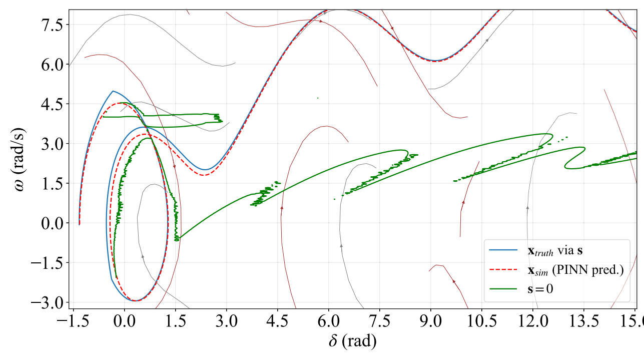


Fig. 6. The $\mathbf{s} = 0$ curve and its relation to the system trajectory

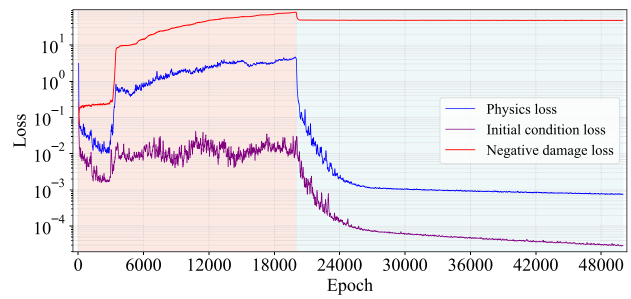
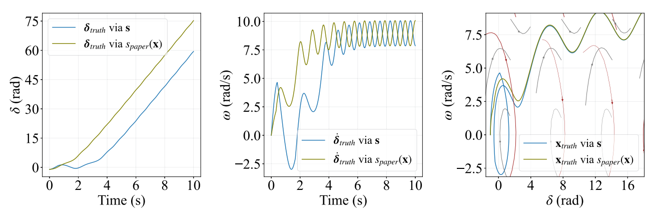


Fig. 7. Log-moving average of the physics loss, initial condition loss, and negative damage loss over training epochs during attack execution



(a) Rotor Angle (δ) (b) Angular Velocity (ω) (c) Phase Portrait

Fig. 8. Physical performance of the attack in [7] with $\mathbf{s}_{paper}(\mathbf{x}_0) < 0$

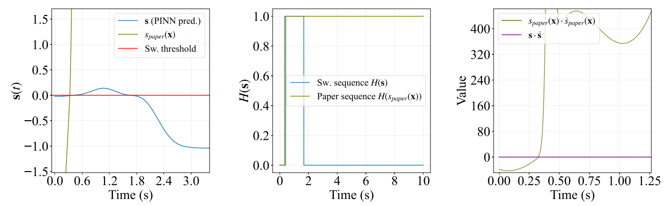
with the sole assumption that the system is initially in a state with the load P_L disconnected, i.e., $\mathbf{s}_{paper}(\mathbf{x}_0) < 0$, which means $S_L(\mathbf{x}) = 0$. Consequently, the initial state $\mathbf{x}_0 = (\delta_0, \omega_0)$ must satisfy $\mathbf{s}_{paper}(\mathbf{x}_0) < 0$, which yields:

$$\mathbf{s}_{paper}(\mathbf{x}_0) < 0 \implies 6\delta_0 + \omega_0 < -4e^{\tanh(6\delta_0 + \omega_0)} \tag{21}$$

To verify the performance of the switching attack proposed in [7], we compare the trained network on system (12) under conditions where inequality (21) either holds or fails.

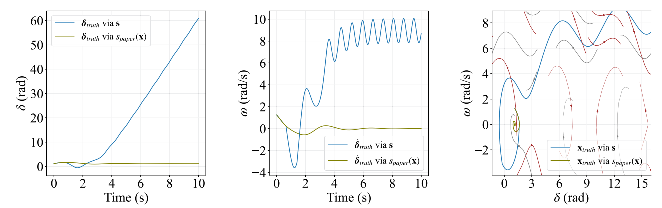
1) $\mathbf{s}_{paper}(\mathbf{x}_0) < 0$: Set the initial state as $\mathbf{x}_0 = (-1.0, 0.0)$, which satisfies the condition: $-6 < -4e^{\tanh(-6)} \approx -1.472$. As shown in Fig. 8, under this condition, the solution consistently outperforms that of the trained network, and the discovered attack exhibits a one-cycle delay in execution. Fig. 9 shows that even after attempts to tune the model, no better attack strategy was automatically discovered.

2) $\mathbf{s}_{paper}(\mathbf{x}_0) > 0$: There exist initial states, such as $\mathbf{x}_0 = (1.15, 1.25)$ with $s_{\text{paper}}(\mathbf{x}_0) > 0$, for which the method pro-



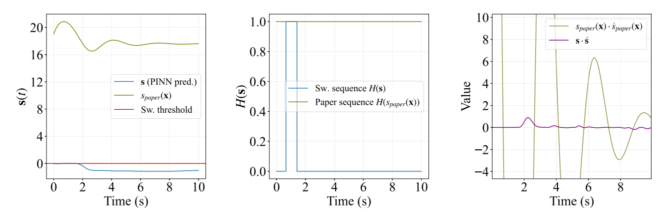
(a) Switching Function (b) Switching sequence (c) Attractivity condition

Fig. 9. Switching state of the attack in [7] with $\mathbf{s}_{paper}(\mathbf{x}_0) < 0$



(a) Rotor Angle (δ) (b) Angular Velocity (ω) (c) Phase Portrait

Fig. 10. Physical performance of the attack in [7] with $\mathbf{s}_{paper}(\mathbf{x}_0) > 0$



(a) Switching Function (b) Switching sequence (c) Attractivity condition

Fig. 11. Switching state of the attack in [7] with $\mathbf{s}_{paper}(\mathbf{x}_0) > 0$

posed in [7] fails to find the attack, as demonstrated in Fig. 10 and Fig. 11. These show that the switching function $\mathbf{s}_{paper}(\mathbf{x})$ is not a universal solution and can still fail to uncover certain attacks. In contrast, the learned approach finds and executes the same attack, leading to divergence.

V. Conclusions and Future Works

This paper presents an automated approach for designing switching attacks in power systems by leveraging PINN. Specifically, the PINN learns the dynamics of the target system while incorporating both attractivity and damage terms into the loss function; this dual-objective optimization facilitates the discovery of valid yet high-impact attack sequences. Simulation results on a SMIB model validate the effectiveness of the proposed method; future research will focus on extending this framework to multi-switch scenarios within higher-dimensional, multi-bus networks.

Acknowledgment

Artificial intelligence tools (Manus AI, ChatGPT, and DeepSeek) were used to assist with editing code and text.

References

[1] S. Su, C. Wu, J. Ma, and X. Zeng, "Attacker's perspective based analysis on cyber attack mode to cyber-physical system," *Dianwang Jishu/Power System Technology*, vol. 38, pp. 3115–3120, Nov. 2014.

[2] M. Swearingen, S. Brunasso, J. Weiss, and D. Huber, "What you need to know (and don't) about the aurora vulnerability," *Power*, vol. 157, pp. 52–+, Sep. 2013.

[3] S. Liu, Y. Feng, D. Kundur, T. Zourntos, and R. K. Butler, "A class of cyber-physical switching attacks for power system disruption," in *Proceedings of the Seventh Annual Workshop on Cyber Security and Information Intelligence Research*, ACM, 2011, pp. 1–4.

[4] S. Liu, S. Mashayekh, D. Kundur, T. Zourntos, and K. L. Butler-Purry, "A smart grid vulnerability analysis framework for coordinated variable structure switching attacks," in *2012 IEEE Power and Energy Society General Meeting*, 2012, pp. 1–6.

[5] S. Liu, S. Mashayekh, D. Kundur, T. Zourntos, and K. Butler-Purry, "A framework for modeling cyber-physical switching attacks in smart grid," *IEEE Transactions on Emerging Topics in Computing*, vol. 1, no. 2, pp. 273–285, 2013.

[6] A. K. Farraj, E. M. Hammad, D. Kundur, and K. L. Butler-Purry, "Practical limitations of sliding-mode switching attacks on smart grid systems," in *2014 IEEE PES General Meeting — Conference & Exposition*, 2014, pp. 1–5.

[7] A. Patel and S. Purwar, "Switching attacks on smart grid using non-linear sliding surface," *IET Cyber-Physical Systems: Theory & Applications*, vol. 4, no. 4, pp. 382–392, 2019.

[8] F. Alanazi, J. Kim, and E. Cotilla-Sanchez, "Load oscillating attacks of smart grids: Vulnerability analysis," *IEEE Access*, vol. 11, pp. 36 538–36 549, 2023.

[9] A. Abazari, K. Sarieddine, M. Ghafouri, D. Jafarigiv, R. Atallah, and C. Assi, "Electric vehicle switching attacks against subsynchronous stability of power systems," *IEEE Transactions on Industrial Informatics*, vol. 21, no. 1, pp. 475–486, 2025.

[10] M. Ghafouri, E. Kabir, B. Moussa, and C. Assi, "Coordinated charging and discharging of electric vehicles: A new class of switching attacks," *ACM Trans. Cyber-Phys. Syst.*, vol. 6, no. 3, Sep. 2022.

[11] M. Raissi, P. Perdikaris, and G. Karniadakis, "Physics-informed neural networks: A deep learning framework for solving forward and inverse problems involving nonlinear partial differential equations," *Journal of Computational Physics*, vol. 378, pp. 686–707, 2019.

[12] V. Sitzmann, J. Martel, A. Bergman, D. Lindell, and G. Wetzstein, "Implicit neural representations with periodic activation functions," in *Advances in Neural Information Processing Systems*, vol. 33, Jun. 2020, pp. 7462–7473.

[13] G. S. Misyris, A. Venzke, and S. Chatzivasileiadis, "Physics-informed neural networks for power systems," in *2020 IEEE Power & Energy Society General Meeting (PESGM)*, 2020, pp. 1–5.